NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

THESIS

AN EVALUATION OF YUMA PROVING GROUNDS BALLISTIC ARSENAL SCORING METHODS

By

Jon S. Francisco, Jr.
Kristopher E. von Krueger

June 2005

Thesis Advisor: Lyn R. Whitaker
Second Reader: Samuel E. Buttrey

Approved for public release; distribution is unlimited.

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE			*Form Approved OMB No. 0704-0188*
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instruction, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188) Washington DC 20503.			
1. AGENCY USE ONLY *(Leave blank)*	**2. REPORT DATE** June 2005	**3. REPORT TYPE AND DATES COVERED** Master's Thesis	
4. TITLE AND SUBTITLE: An Evaluation of Yuma Proving Grounds Ballistic Arsenal Scoring Methods			**5. FUNDING NUMBERS**
6. AUTHORS **Francisco, Jon S.** **Von Krueger, Kristopher E.**			
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey, CA 93943-5000			**8. PERFORMING ORGANIZATION REPORT NUMBER**
9. SPONSORING /MONITORING AGENCY NAME(S) AND ADDRESS(ES) N/A			**10. SPONSORING/MONITORING AGENCY REPORT NUMBER**
11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.			
12a. DISTRIBUTION / AVAILABILITY STATEMENT Approved for public release; distribution is unlimited.			**12b. DISTRIBUTION CODE**

13. ABSTRACT

Yuma Test Center (YTC) is a primary testing and evaluation facility for the United States Army's 30mm M230 automatic gun. The current program used by YTC to evaluate 30mm rounds is an old, complex, FORTRAN package called Accuracy Scoring Program (ASCORE). Due to the age and complexity of ASCORE, questions about its reliability in ballistic calculations have been raised. These reliability questions are important, because ASCORE is used by YTC to evaluate whether all such weapons systems meet Department of Defense specifications. Currently, the Army has no plans to replace ASCORE. The Navy has expressed interest in using ASCORE for fire control We evaluate ASCORE using a modern ballistic simulator named Projectile Design and Analysis System (PRODAS) to simulate trajectories of live fire rounds. This thesis demonstrates how PRODAS can be used to evaluate ASCORE. Preliminary results indicate that ASCORE's accuracy decreases the greater a round's impact distance is from the target.

14. SUBJECT TERMS Virtual Target, ASCORE, PRODAS, Ballistic Trajectory			**15. NUMBER OF PAGES** 135
			16. PRICE CODE
17. SECURITY CLASSIFICATION OF REPORT Unclassified	**18. SECURITY CLASSIFICATION OF THIS PAGE** Unclassified	**19. SECURITY CLASSIFICATION OF ABSTRACT** Unclassified	**20. LIMITATION OF ABSTRACT** UL

NSN 7540-01-280-5500 Standard Form 298 (Rev. 2-89) Prescribed by ANSI Std. 239-18

THIS PAGE INTENTIONALLY LEFT BLANK

AN EVALUATION OF YUMA PROVING GROUNS BALLISTIC ARSENAL SCORING METHODS

Jon S. Francisco, Jr.
Ensign, United States Navy
B.S., United States Naval Academy, 2004

Kristopher E. von Krueger
Ensign, United States Navy
B.S., United States Naval Academy, 2004

Submitted in partial fulfillment of the
requirements for the degree of

**MASTER OF SCIENCE IN APPLIED SCIENCE
(OPERATIONS RESEARCH)**

from the

**NAVAL POSTGRADUATE SCHOOL
June 2005**

Author: Jon S. Francisco, Jr.

Kristopher E. von Krueger

Approved by: Lyn R. Whitaker
Thesis Advisor

Samuel E. Buttrey
Second Reader

James N. Eagle
Chairman, Department of Operations Research

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

Yuma Test Center (YTC) is a primary testing and evaluation facility for the United States Army's 30mm M230 automatic gun. The current program used by YTC to evaluate 30mm rounds is an old, complex, FORTRAN package called Accuracy Scoring Program (ASCORE). Due to the age and complexity of ASCORE, questions about its reliability in ballistic calculations have been raised. These reliability questions are important, because ASCORE is used by YTC to evaluate whether all such weapons systems meet Department of Defense specifications. Currently, the Army has no plans to replace ASCORE. The Navy has expressed interest in using ASCORE for fire control We evaluate ASCORE using a modern ballistic simulator named Projectile Design and Analysis System (PRODAS) to simulate trajectories of live fire rounds. This thesis demonstrates how PRODAS can be used to evaluate ASCORE. Preliminary results indicate that ASCORE's accuracy decreases the greater a round's impact distance is from the target.

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

LIST OF FIGURES

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF TABLES

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF ABBREVATIONS

ASCORE	Accuracy Scoring
EA2G	Enhanced Air to Ground
IRCC	Inverted Range Coordinate Center
NPS	Naval Postgraduate School
OHS	Overhead Score
PRODAS	Projectile Design Analysis
SPM	Shots Per Minute
YTC	Yuma Test Center
VT	Virtual Target

THIS PAGE INTENTIONALLY LEFT BLANK

ACKNOWLEDGMENTS

The authors would like to thank their families and friends for their unending support. A special thanks to Professor Lyn Whitaker for advising on this thesis. Your support and guidance was a great help and will not be forgotten. We would also like to acknowledge the efforts of Mike Diehl, Barbara Carlson, John Curry, and all the support staff at the Yuma Test Center for their assistance. Additional thanks to Professor Samuel Buttrey for second reading, and to Professors Robert Koyak and David Annis for their time, support and guidance.

THIS PAGE INTENTIONALLY LEFT BLANK

EXECUTIVE SUMMARY

One of the United States Army's current weapons testing and evaluation facility is located at Yuma Test Center (YTC). YTC evaluates the firings of 30mm rounds from the M230 automatic gun attached to the bottom of the AH-64 Apache helicopter. The current program used to evaluate these fired rounds is the Accuracy Scoring Program (ASCORE) which is old and complex. There have been questions raised about ASCORE's accuracy in its ballistics calculations. This is an important issue because the Army uses results from YTC to ensure its weapons systems meet specifications. Currently, there is no replacement program for ASCORE and the Army plans to continue to use it in future tests. We evaluate ASCORE by using a more modern ballistic simulator called Projectile Design and Analysis System (PRODAS).

ASCORE is a legacy, LINUX based program that was originally written in FORTRAN. It has been modified and translated into different computer languages numerous times. There is currently no engineer at YTC who completely understands the inner workings of ASCORE. This has lead YTC to believe that ASCORE has possibly become outdated and that many of the algorithms used for ballistic calculations may be performing badly compared to modern ballistics software.

Currently ASCORE is used to pass the 30mm M230 gun, among other weapons systems, specifications. YTC uses ASCORE to calculate a virtual target impact. The virtual target is an imaginary plane perpendicular to the line of sight of the aircraft and passing through the real target center. It is the virtual target impact that indicates whether or not a weapon is hitting its target within specifications. It is imperative that the Army use programs that perform well so that the best weapons systems can be available for military operations. To date, there has been no attempt to replace ASCORE with a more modern ballistic simulator. In fact, the U.S. Navy has expressed interest in using ASCORE as part of its fire control. This fact makes an evaluation of ASCORE even more pressing.

In this thesis, we use PRODAS to simulate trajectories of ten-round bursts of 30 mm rounds fired from a hovering Apache Helicopter. We treat these simulated trajectories as YTC does a real test firing. Their virtual target (VT) impact locations are approximated by ASCORE. We evaluate ASCORE accuracy by comparing these approximate VT impact locations to the “actual” VT impact location of the trajectories simulated by PRODAS.

This thesis demonstrates how PRODAS can be used to evaluate ASCORE. Details for running both ASCORE and PRODAS as well as all code needed to pass trajectory data from one program to another are given in the appendixes. Preliminary experimentation shows that when the range to the target is less than 750 meters, ASCORE accuracy suffers. In addition, there is evidence that ASCORE accuracy decreases as miss distance increases. It will be important for YTC to investigate this further because miss distances typically observed by YTC are greater than those studied in this thesis.

I. INTRODUCTION

The Yuma Test Center (YTC), located in Yuma AZ, is one of the Army’s primary weapons testing and evaluation facilities. Currently, YTC is working with the Naval Postgraduate School’s (NPS) Operations Research Department to resolve various issues concerning weapons accuracy and precision. The official name given to this YTC program is The Enhanced Air to Ground program (EA2G).

The EA2G program involves live fire tests designed to develop improved methods for scoring air-to-ground ballistic arsenal engagements. “Scoring a round” is the assignment of impact coordinates to a round. By improving the YTC ability to accurately evaluate live fire tests, development and validation of future Advanced Attack Helicopter weapons systems can be reached more quickly and at a reduced cost. Scoring these live fire tests involves identifying the impact location of each projectile fired. This scoring is done manually with the aid of video cameras and then evaluated by an Accuracy Scoring program (ASCORE). The output of ASCORE is interpreted as the final result of a test firing. Consequently, much of the information about accuracy and precision of weapons performance comes from the interpretation of ASCORE’s output. ASCORE is a legacy program originally written in FORTRAN that has been modified and translated so many times that no engineer currently at YTC completely understands its inner workings. This has lead YTC to question ASCORE’s accuracy. In fact, some suspect its algorithms may be performing poorly compared to more modern ballistic algorithms. This thesis focuses on evaluating the ASCORE program based on 30mm rounds fired from a hovering helicopter.

A. TEST FIRING

The Army’s M230 automatic gun, Figure 1.1, is designed to fire 30mm linkless ammunition at a rate of about 625 shots per minute (SPM). The M230 is a component of the area weapon system of the AH-64A Apache Helicopter. The M230 is a single barrel, externally powered, electrically fired, chain driven weapon.

Figure 1.1. Pictures of the AH64A Apache Helicopter and the M230 automatic gun.

When a test firing is performed, the M230 uses an auto tracking system to lock in on a 10ft x10ft vertical white target. Since the tracking system of the Apache focuses on the white of the target, the target is repainted several times throughout a test firing to ensure consistency. Two cameras are synchronized prior to a test firing and record the impacts of the 30mm rounds on and around the target. The first camera is on the ground, immediately in front of the target, and can easily identify direct hits to the target. However, this vantage point provides little information on the impact time and location of rounds that miss the target. The second camera resolves this issue; it is mounted on a helicopter flying well above the firing range and records a “sky view” of the test firing. These two films are later reviewed by a YTC analyst to establish the time and location of impact for each round that is fired. This is called the Overhead Score (OHS). The location of the target and firing positions of the helicopter are clearly marked on an area map for reference in Appendix A

1. Overhead Scoring

The OHS is computed by a YTC analyst who reviews the film of the test firing taken by the sky camera. The sky camera film is reviewed frame by frame and each "disturbance" in the target area is examined to determine if an impact occurred. The general rule used when deciding if an impact has occurred is, "if in doubt, score it as an impact." This results in almost all "disturbances" being marked as an impact. As can be imagined, the OHS is often impaired by multiple factors which include:

- Environmental effects (The desert test site can stir up a sizable amount of dust. This can mask possible impacts from the camera or make them very difficult to identify on film.)
- Sand skipping (The surface of the test site has an initial hard cover that can occasionally cause a round that impacts at a shallow angle to skip. This often appears as two independent impacts during the OHS review.)
- Human error (Even though the scorer is a very experienced and highly skilled technician, some human error is introduced.)
- Video resolution (The film that is reviewed by hand to compute the OHS is done so frame by frame. One frame represents 16.7 milliseconds. Due to the high speed of the 30mm round, this speed of resolution is too slow to correctly differentiate whether one or multiple impacts have occurred.)

Approximately eighty percent of all rounds fired in the typical ten-round burst are found in the film review and scored. The rest of the fired rounds are either lost or miss scored usually due to one of the reasons listed above. The accuracy of the OHS is very important because the resulting data is fed into ASCORE for analysis. Based on the OHS inputs, ASCORE makes two important calculations involving round matching and the virtual target impact (VT).

2. Round Matching

Before processing a burst, ASCORE takes as input how many rounds were fired in the burst. It uses this information combined with the weapons SPM to calculate an approximate fire time for each round that was supposed to leave the gun. ASCORE must then match each of these fire times with the most likely scored impact location and time (taken from the OHS data). This process is quite involved. Not only is it possible that a recorded impact is not actually an impact, but rounds tend to pass each other in the air depending on their initial trajectory. Thus, the round leaving the gun first may not be the first to impact. ASCORE uses an internal round matching algorithm to sort through the data and pair impact times to fire times. One area of concern with this algorithm is that it is very old and none of its creators are available. Since it is nearly impossible to recover the shots fired from the range, there is currently no good way to evaluate how well the round matching algorithm truly works.

3. Virtual Target

The purpose of scoring the impact of a round is to determine if the weapon fired is accurate enough to meet weapons specifications. Specifications are written for targets of specified sizes which are perpendicular to the line of fire. Unfortunately, scoring of the rounds that impact the target is complicated by the fact that true angle of fire is never perpendicular to the target. Thus the surface area of the target presented to the gun can actually be much less than the specified area of the target. This results in rounds missing the target that would have normally hit had the target been perpendicular to the line of fire. To determine if this has occurred, the location of a virtual target (VT) is calculated for each round fired. The VT is a plane that pivots on the center of the real target. This plane is rotated in such a way that it is perpendicular to the gun's line of fire for each shot. The virtual target impact is the point at which the trajectory of a round intersects the virtual target. Since this often happens in midair or after the round has hit the ground, a back trajectory or forward trajectory must be computed based on the initial firing parameters (time and location) and the ground impact parameters for each round which misses the actual target. ASCORE approximates the trajectory based on the fall angle of the projectile at the time of impact and initial distance from the target. This trajectory is used to calculate the VT impact location. The fall angle is the angle at which the projectile intercepts its real impact point. One interesting aspect of this approximation is that ASCORE uses two different algorithms to calculate the fall angle based on the initial target range at the time of fire. If the range is less then 750 meters then one type of approximation is used. If the initial target distance is greater than 750 meters than another approximation is used. As with the round-matching algorithm, the VT computations in ASCORE are very old and the details of the computations and not well understood by those at YTC.

B. EVALUATING ASCORE

The primary goal of this thesis is to evaluate how accurate the ASCORE virtual target calculation and round-matching algorithms are over a range of conditions for 30mm projectile trajectories. Because is not possible to observe trajectories and actual impact locations of live fired rounds, 30mm round trajectories are simulated using a ballistics program called “Projectile Design and Analysis,” or PRODAS, that uses more modern algorithms than ASCORE. It is critical that PRODAS be able to generate a sufficient number of trajectories under varying conditions typical of those found at YTC. (This is accomplished by using a preconfigured 30mm design within PRODAS in conjunction with its projectile simulation module). By using PRODAS to simulate multiple test firings, the parameters, conditions, and trajectories of each can be known. With this knowledge it is possible to evaluate the effectiveness of ASCORE.

To generate “true” trajectories from PRODAS and then evaluate their impacts using ASCORE, it is necessary to understand both ASCORE and PRODAS and design a number of routines allowing the two programs to interact. This presents significant challenges as ASCORE is a LINUX-based program translated into C and PRODAS is a Windows-based program operating on Visual Basic scripts. Thorough documentation of ASCORE, PRODAS, and all additional programming used in this thesis are included in the appendixes.

C. THESIS OUTLINE

Chapter II gives a brief description of general ballistics and defines the basic variables that can affect 30mm round trajectories and therefore apply to this thesis. More specific detail on the workings of ASCORE and PRODAS is also given. Chapter III explains the process used to generate the data used for the analysis. The experimental design used in this thesis is also given, including the specific assumptions and ranges for the variables used. The chapter concludes with a preliminary analysis. Chapter IV contains the primary analysis. All statistics and resulting analysis are included in this chapter. Chapter V concludes with a summary of results, a list of problems encountered and specific areas where future research is required.

II. BACKGROUND

A. BALLISTICS

While the scope of this thesis is focused on evaluating ASCORE based on 30mm rounds, a brief description on general ballistics is necessary before proceeding. Ballistics is the study of the processes within a firearm as it is fires, and the science of the motion of projectiles in flight [Ref. 1]. From this definition three aspect of ballistics are derived: internal, terminal, and external.

Internal ballistics refers to what happens inside the weapon from the instant of fire until the round exits the muzzle. This includes combustion, pressure development and motion of the round along the bore of the firearm [Ref. 2]. Because we are not developing new types of rounds, but are using existing 30 mm rounds, this thesis will focus on the events of a round outside a weapon.

Terminal ballistics begins the instant a round enters a target, and how it behaves once entering the target. As expected, different rounds behave differently. For example, some rounds penetrate the target while others explode upon impact [Ref. 2]. Terminal ballistics are important when studying the lethality of a round, but is not in the scope of this thesis.

External ballistics describes what occurs to the round from the moment it exits the muzzle until it impacts the target. There are a wide variety of factors that can affect a round during this time period. Meteorological conditions, air friction, the earth's rotation, muzzle velocity, and drift are some of these factors [Ref. 2] and [Ref. 3]. External ballistics is our primary focus.

1. Mass and Aerodynamic Forces

Mass forces refer to those forces that apply at the center of gravity of the round and depend on the body mass and mass distribution of the round. Gravity, Coriolis, and centrifugal force are grouped together as mass forces. Aerodynamic forces are a result of the interaction of airflow with the round and depend on the shape and smoothness of the round. Drag, lift, magnus, pitch damping, and transversal magnus are grouped together into aerodynamic forces. For a more in-depth explanation of these forces and their properties refer to [Ref. 4].

2. Round Stability

A round in a stable state is one whose longitudinal axis is coincident with the direction of movement. A stable round must satisfy three conditions: static stability, dynamic stability, and tractability.

Yaw angle, wind force and overturning moment must be understood in order to define the various conditions of stability. A round with a yaw angle is one for which the direction of motion of the round's center of gravity deviates from the direction of the round's axis of symmetry. Wind force occurs in rounds with a yaw angle and is the force that results from pressure difference at the round's surface. If there are two wind forces then this pair is a free vector which is known as the overturning moment. The overturning moment attempts to rotate the bullet around an axis [Ref. 4].

Static stability occurs when a round responds to a wind force by moving its nose into the direction of the overturning moment. If the round does not have static stability then the overturning moment will cause the bullet to tumble. A round possesses dynamic stability if the the yaw angle decreases over time. If the round is dynamically unstable the yaw angle increases over time. All projectiles have a yaw angle which is introduced at the muzzle. The initial yaw angle as the round leaves the muzzle is not an indication of a round's dynamic stability. A round's tractability describes the rounds ability to let its longitudinal axis follow a bending trajectory. A round may cease to be tractable if it is over-stabilized. A round becoming over-stabilized is more common with high-angle shooting. Over-stabilized occurs when that the round is rotating too fast and becomes unable to follow a bending trajectory (see Figure 2.1) [Ref. 4].

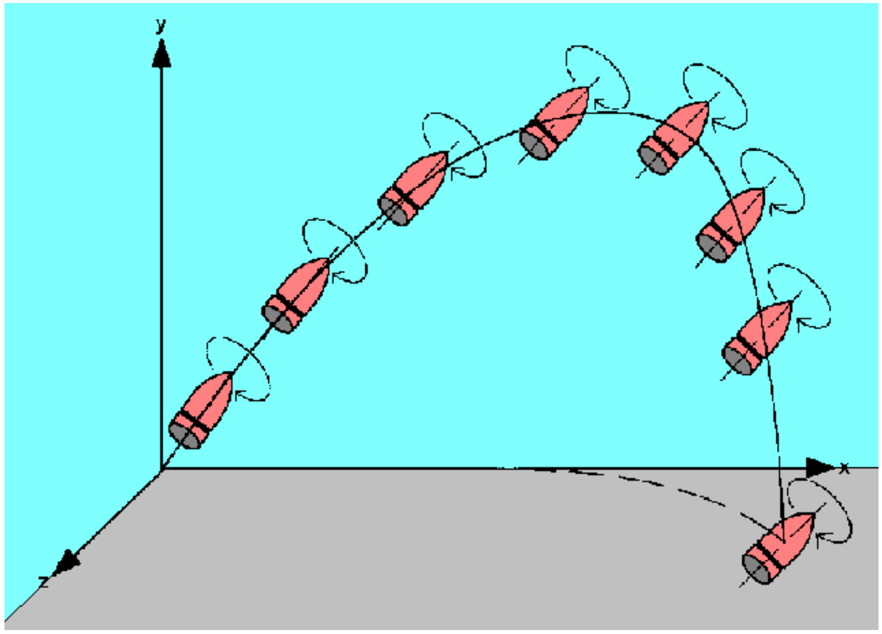


Figure 2.1. An over-stabilized bullet on a high-angle trajectory [Ref. 4].

3. Effects of External Ballistics

In this section, we discuss many of the factors that influence a round's path. They include muzzle velocity, aerodynamic drag, the earth's rotation, drift, and various meteorological conditions. Muzzle velocity is defined as the speed of a projectile as it leaves the muzzle of a weapon [Ref. 5]. When a firing is parallel to the ground, faster muzzle velocity induces a flatter trajectory. Conversely, lower muzzle velocity causes a loss in altitude. In addition, higher temperature corresponds to increased muzzle velocity. This must be considered since the rounds heat up as the number of shots fired from the same weapon increases [Ref. 3]. A measure of drag on a round is known as the ballistic coefficient. This causes an increase in the vertical drop of the bullet away from its original line of departure and also a decrease in the round's impact velocity [Ref. 3].

The earth's rotation causes what is knows as the Coriolis Effect on a round's path. The Coriolis Effect is the movement of the target, due to the earth's rotation, from the time of fire of the projectile. The Coriolis Effect for short range fire is negligible but in long range firings (namely artillery fire) it non-negligible. The magnitude and direction of the Coriolis Effect depends on the latitude and orientation of the location where the round is fired. The direction of the Coriolis Effect depends on the direction the round is being fired. Accounting for the Coriolis Effect is similar to leading a moving target in small range firings [Ref. 3].

Gyroscopic precession causes drift in the trajectories and must be accounted for in all spin stabilized projectiles. Drift is more severe for projectiles fired over long ranges than for projectiles fired over short ranges [Ref. 3].

Meteorological conditions influence a projectile's trajectory. Wind speed and direction are two such factors. A round's horizontal direction will change due to wind. Wind speed and direction can be measured using appropriate meteorological instruments and aiming adjustments must be made by either the gunner or a computer that aims the weapon. Furthermore barometric pressure (which depends on altitude) is an indicator of air density, which affects round trajectories. Denser air causes an increase in aerodynamic drag. Air temperature affects the aerodynamic drag of a round in flight, as it affects both density and velocity. Therefore, a supersonic round's velocity will be lower in colder air. Finally, relative humidity, similar to barometric pressure, affects the density of the air. Humid air is less dense than dry air having the same conditions of barometric pressure and temperature. Therefore more humidity means less aerodynamic drag [Ref. 3].

4. Experimental Parameters

Ideally, this thesis would evaluate ASCORE ballistics based on PRODAS's simulated trajectories by varying all possible factors that can affect a round's trajectory. However, this is not possible since all of these factors are either non-exsistant or not controllable in both ASCORE and PRODAS. Aerodynamic drag, drift and the earth's rotation are not controllable in either PRODAS or ASCORE, and therefore are ignored as variables in this thesis. Since this thesis compares the PRODAS virtual target impact to the corresponding ASCORE estimate, we are in a sense comparing how closely each program computes the drag and drift and takes into account the earth's rotation.

The two parameters varied are air temperature and barometric pressure (air pressure). Both have measurable affects on aerodynamic drag. ASCORE and PRODAS have the means to change both of these parameters for our simulations. Since YTC is in the desert, both air temperature and air pressure can vary dramatically depending on the season and time of day. ASCORE and PRODAS allow the elevation the firing location as an input. However, as air pressure is related to altitude, the measured elevation (relative to sea level) at YTC is used in both programs and not varied throughout the course of the study.

Muzzle velocity is another important parameter that should be varied in an experiment. Unfortunately PRODAS sets the muzzle velocity of the M230 gun to 805m/s and it can not be varied. To match the two programs, the muzzle velocity in ASCORE is set to 805m/s. Relative humidity is another factor ASCORE allows as an input but PRODAS does not. Wind speed and wind direction are both changeable in PRODAS and ASCORE. These two parameters are important to consider because they have a considerable affect on external ballistics.

5. 30mm Ammunition/M230 Automatic Gun

This thesis is concerned with 30mm gunfire from a M230 mounted on an Apache helicopter. The 30mm is used in air-to-ground operations, and has proved effective during Operation Desert Storm where it performed well against all targets, including tanks, armored vehicles and light vehicles. Figure 2.2 is a diagram of a typical 30mm round and Table 2.1 gives a summary of the characteristics of the 30mm round as prescribed by PRODAS V3.

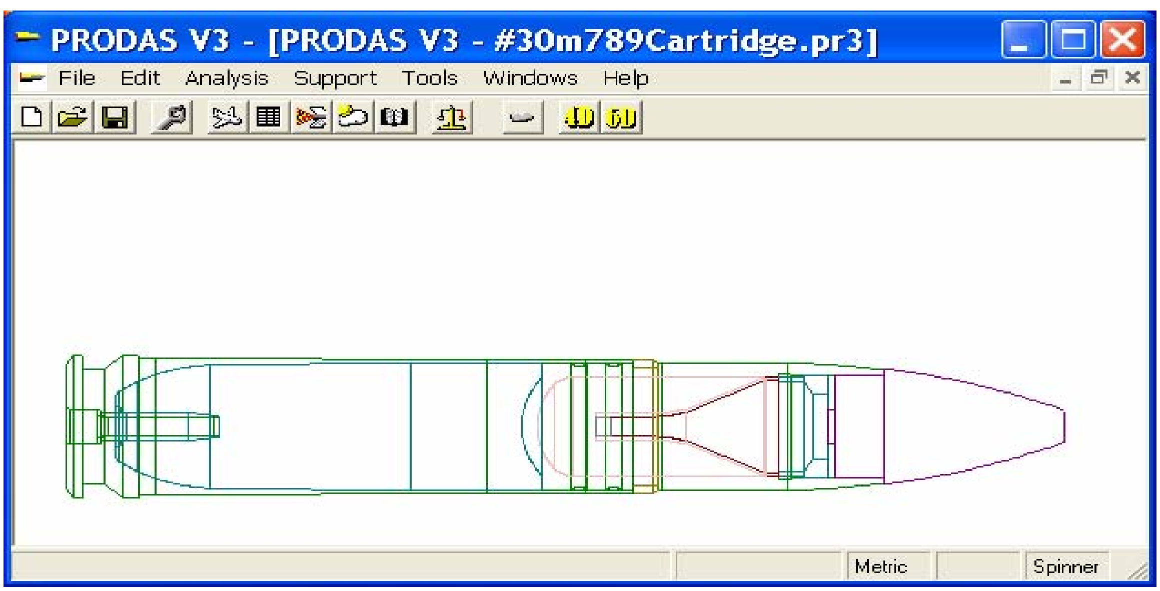


Figure 2.2. Diagram of 30mm M789 (HEDP) High Explosive Dual Purpose Round

#30m789Cartridge.pr3 - 30MM M789 HEDP
06/03/2005 20:13
Document2000 Version 3.0.0

Projectile Length	: 108.1360 mm	* Locked
Ogive Length	: 55.0500 mm	* Locked
Ogive Radius	: 145.0000 mm	* Locked
Meplat Diameter	: 7.5000 mm	* Locked
Boattail Length	: 0.0000 mm	* Locked
Boattail Diameter	: 29.9200 mm	* Locked
Boom Length	: 4.1656 mm	* Locked
Boom Diameter	: 0.0000 mm	
Rotating Band Length	: 5.0790 mm	* Locked
Rotating Band Diameter	: 31.2000 mm	* Locked

	Mass	Transverse Inertia	Axial Inertia	CG from Reference	Density
	gm	gm-cm^2	gm-cm^2	mm	gm/cm^3
Assemblies:					
Projectile	227.3830	1572.600	307.4060	41.55480	
Cartridge C	58.39630	584.1170	103.8550	30.06270	
Total	337.7040	8698.311	467.1680	102.6840	
Components:					
Body	36.9720	668.7130	230.5870	33.62990	7.833411
Retainer	11.97160	7.300140	12.94150	6.128970	7.832971
Liner	8.686200	11.69750	5.400230	23.33690	8.299970
Nose Fuze	38.82290	59.46300	25.16810	18.67380	2.499990
Band	7.118190	7.850890	15.41010	2.465430	8.802171
PBXN-5	23.72360	36.85650	17.89710	20.16250	1.649990
Detonator	0.88481E-01	0.18850E-02	0.24430E-02	1.500000	1.699990
Al Cartridg	55.60270	577.7340	103.6440	30.71470	2.796000
Flashtube	2.793620	1.441310	0.2112360	11.04020	7.833000
Propellant	51.92420	293.6300	55.90680	54.95770	0.9800000

Table 2.1. Summary of 30mm M789 HEDP characteristics

The M230 is manufactured by Boeing and McDonnell Douglas. Its characteristics are summarized in Table 2.2 as an output from PRODAS V3.

Gun Name	M230 Lt Wt 30	
Chamber Volume	0.000	m^3
Barrel Length	0.954	M
Gun Barrel Bore	0.030	M
Bore Grove Diameter	0.031	M
Rifling Depth	0.000	M
Groove Land Ratio	2.277	
Number Land Groove	16.000	
Start Angle	0.000	radian
End Angle	0.114	radian
Twist	27.570	cal/rev
Projectile Free Travel	0.000	M
Forcing Cone Half Angle	0.17453	radian

Table 2.2. Summary of M230 30mm automatic gun characteristics

B. PRODAS

PRODAS V3 was developed by Arrow Tech Associates Incorporated. The main purpose of PRODAS is to perform rapid evaluation on the performance of ammunition characteristics. To this end, PRODAS has the capability to design and evaluate projectiles. The former is not a concern in this thesis. PRODAS is a deterministic ballistic model that computes the trajectory of one round at a time. PRODAS conducts several different analyses which it links together into a common database so that each subsequent analysis can utilize the results of a prior analysis. Currently YTC does not use any PRODAS analyses. Nor does it use PRODAS to simulate firings. Refer to Appendix B for a more in-depth explanation of PRODAS.

1. Analyses and Parameters

The first step in performing firing simulation is to determine the round that will be fired in the simulator. Once this has been decided, PRODAS provides a number of analyses, including those concerning mass properties, aerodynamics, dispersion, interior ballistic and trajectories. More information can be found on each by referring to the PRODAS User Manuel [Ref. 6].

Once a projectile has been created in the PRODAS ammunition database, its design can be used repeatedly without having to recreate the round. We used the 30mm round, which was pre-programmed into this database. PRODAS does have the capability to change many input parameters, such as airplane velocity, altitude, bank angle, and dive angle, initial location, quadrant elevation, gun azimuth, and meteorological conditions (Refer to Appendix B for a more detailed description of these variables).

2. Visual Basic Script

Users may perform unattended analysis by PRODAS via Visual Basic (VB) script. This allows the user to open files describing a projectile, change parameter values, run analyses and post process. Similarly, a script can also execute a sequence of PRODAS runs. This aspect is extremely valuable to this study due to the large number (700) of simulated rounds fired.

3. Uses for PRODAS

This thesis is not concerned with PRODAS's ability to design projectiles. For this study, PRODAS is solely used to simulate test firings of 30mm rounds. More specifically, the goal is to simulate shots in PRODAS and input the results of those "shots" into ASCORE. In order to do this, the results of PRODAS must provide the same information as OHS does for live fired rounds. To do this, the 30mm trajectory analysis, analysis of the mass properties, and the aerodynamics analysis are needed. Instead of executing all three analyses each time a single round is fired, a script was created for each round for a total of 700 scripts. As each script is run in PRODAS a corresponding results text file is created, containing all the relevant information needed by ASCORE.

C. ASCORE

ASCORE is the ballistics program currently used by YTC for approximating the ballistic trajectories, computing the corresponding virtual target, and compiling the results of range firings. ASCORE is a FORTRAN-based legacy program and much of the programming and models implemented in the code are not well understood. ASCORE evaluates the accuracy of fired munitions by matching pulses to impacts. The program takes as input the parameters of the specific firings, such as ammunition type, target location, meteorological conditions, helicopter position and velocity, impact location, and pulse time. ASCORE matches each round fired with the most likely corresponding pulse using a hard-coded algorithm. Usually rounds are entered into ASCORE as either ten or twenty-round bursts. For this thesis, ten-round bursts are considered. Refer to Appendix C for a more in-depth explanation of ASCORE.

1. Input Files

ASCORE uses several different files as inputs for a burst of shots.

- Ascore.run – A large number of parameters needed to run ASCORE are in this file. (This includes for example the location of other input files used for the burst and the coordinates of the target and which meterological towers are used.)
- Flt4.tv – data for the impact of the fired rounds. The impact location coordinates are relative to the center of the target.
- Rte-223.ssf – data on the position of the helicopter. The coordinates for the location of the helicopter are relative to a location on the TYC firing range called the Inverted Range Coordinate Center (IRCC) (See Appendix A for a map of the YTC firing range).
- Mettower08.ssf - Meteorological parameters are entered into ASCORE through three different files named Mettower08.sff, Mettower10.sff, and Mettower12.sff. Each meteorological tower needs data from each of four levels on the tower. ASCORE takes the data from each level on all three towers and averages them.

2. Uses for ASCORE

In this study, ASCORE's internal workings are immaterial. Rather, we use ASCORE as YTC would and compare the results to known trajectories. Information from the PRODAS simulations are extracted and input in to ASCORE. Several issues during this process require special attention.

PRODAS and ASCORE use different coordinate systems. To match the coordinate systems PRODAS must switch the values of its X and Y coordinates and multiply the new x-coordinate by negative one (see Figure 2.3). Both programs use the center of the target as the coordinate origin. Determining the location of the helicopter requires a linear transformation since ASCORE requires this parameter to be entered relative to the IRCC .The PRODAS output of the helicopter position is relative to the center of the target. PRODAS also only gives as an output one set of meteorological conditions for each shot. To input this data in ASCORE, the same values for all four levels of the meteorological tower are entered as the meteorological conditions for each of the three towers.

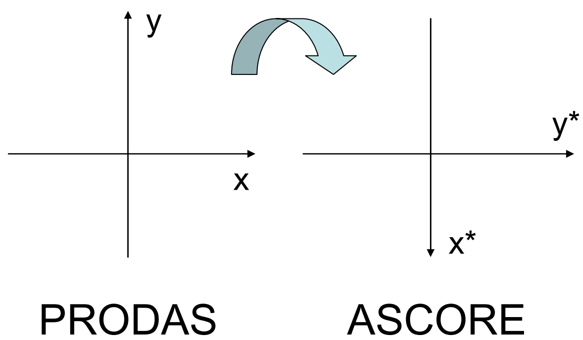


Figure 2.3. Matching PRODAS coordinates to ASCORE.

III. CONDUCTING THE EXPERIMENT

A. EXPERIMENTAL FLOW

The process of generating experimental data for this thesis is a detailed process. An overview of the process is given in Figure 3.1.

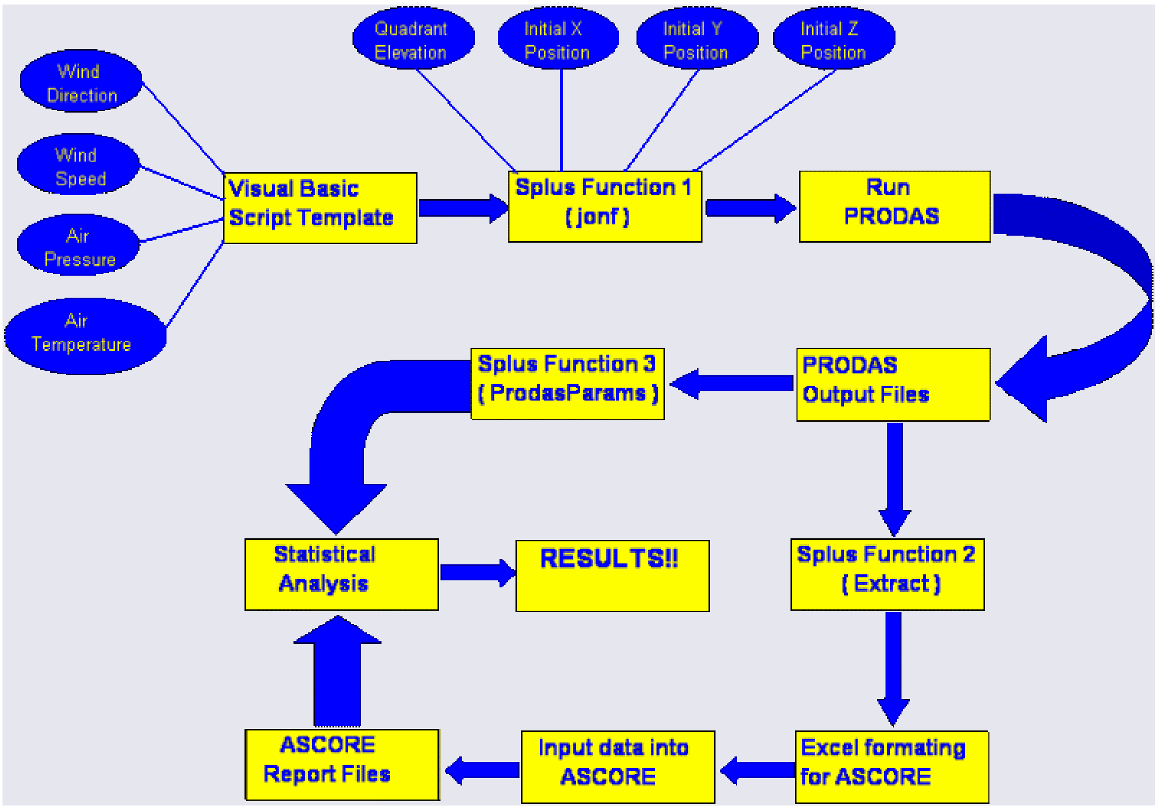


Figure 3.1. Flow chart for the experimental data generation.

Before generating data, a decision must be made as to which variables to vary. By analyzing the input parameters for both PRODAS and ASCORE, a list of variables that can be modified by each program can be obtained; this list is shown in Table 3.1.

PRODAS Variables	ASCORE Variables
Wind Speed Wind Direction Helicopter (X,Y,Z) Position Air Pressure Temperature Quadrant Elevation	Wind Speed Wind Direction Helicopter (X,Y,Z) Position Air Pressure Temperature relative humidity

Table 3.1. Table of common PRODAS and ASCORE variables of interest.

Unfortunately, major difficulties were encountered when attempting to modify the data file responsible for reporting the wind speed and wind direction within the version of ASCORE currently in the possession of NPS. After conversing with YTC (Barbra Carlson, personal communications, May 10, 2005) about how to best correct this problem, it was decided that fixing the module responsible would have to be done at a later time and would therefore be left for future research studies. For this reason, the wind speed and wind direction were held constant at the default values of 2.0 mph and 2.0 radians respectively.

While PRODAS processes the ballistics for each round independently of all other rounds, ASCORE processes them as a burst containing anywhere from 1 to 50 rounds. Ten-round bursts will be used in this experiment because not only is representative of a typical burst size in a test firing at YTC, but it also provides a sufficient number of shots within each burst to evaluate the ASCORE round matching algorithm.

The temperature and air pressure are held constant for all rounds in each burst. An initial experiment of 50 bursts (500 rounds) is generated to adequately vary temperature, and air pressure among bursts. The quadrant elevation and helicopter position are modified with each round fired. The quadrant elevation is varied with each shot to represent the random movement of the gun during a test firing. The initial (X,Y,Z) location of the helicopter is varied to emulate the random movement of the helicopter about its hovering position when firing. The option to change the relative humidity is currently not available in PRODAS. Therefore a default value of 34% is used for all ASCORE calculations.

The first step in this experiment is to create a Visual Basic script template for ASCORE. An example of this template is shown in Appendix D along with an explanation of how it was created. Each round generated requires its own unique script.

An S-Plus function is written to automate part of this process. Each variable be held constant during each burst (wind speed, wind direction, air pressure, and temperature) is manually entered into the script template. An S-Plus function called “jonF” (Please see Appendix E for a more detailed description of “jonF”) then reads the script template and generates as many scripts as the user specifies with the quadrant elevation and initial location randomly generated for each one. After every ten scripts generated by jonF the script template must be altered to incorporate updated air pressure and temperature for the next burst. This results in running jonF 50 times, generating a total of 500 PRODAS scripts. Unfortunately, each script must be manually run in PRODAS because the ability of PRODAS to automatically run a set of scripts is still under development by the original programmers. After running a script, PRODAS produces an output file for that script containing details on the ballistic trajectory and the values of the variables specified in Table 3.2. Each output file is large containing twenty columns and several thousand rows. To quickly and efficiently extract the required information from this output file the two S-Plus functions “prodasparams” and “extract” are used. The S-Plus function “prodasparams” reads all the output files in a single directory, reads the values of the PRODAS variables listed in table 3.2 and puts them all into a single data frame that can be later used for statistical analysis. The S-Plus function “extract” reads through the trajectory data for each output file and generates a data file containing the following information :

- Whether the round impacted the ground or the real target
- The simulation time of the real impact
- The coordinates of the real impact
- The simulation time of the virtual target impact
- The X and Z coordinates of the virtual target impact

For more information on the details of “prodasparams” and “extract”, please see Appendixes F and G respectively.

Using the data frame extract outputs, the series of files required to run ASCORE can be constructed. For each burst a unique input file must be created for the position of the airplane at the exact time of firing, the real impact time and coordinates for each round, and the meteorological conditions during the burst. These files are created in Excel and then exported to the LINUX system for input into ASCORE as a comma-delimited file (for more detailed information on this process please see Appendix C). Each burst must be processed by ASCORE seperatly. This results in 50 output reports, one for each burst. The report files contain the virtual target impact coordinates and indicate which rounds were successfully matched. The virtual target is reported in terms of the X and Z coordinates: these are called the azimuth and elevation in the ASCORE report (refer to Appendix C for detailed information on azimuth and elevation). An example of the ASCORE report file and a detailed description of how to read it can be found in Appendix C 3.1. The next step is to export the report files back to a Windows system and translate them into a ASCII file. The data containing the virtual target and round matching can then be extracted and added to the “pradasparams” data frame. With this data frame a method for comparing the PRODAS and ASCORE virtual target impacts and evaluating the ASCORE round matching algorithm can be developed.

B. EXPERIMENTAL DESIGN

The first decision in determining the experimental design is selecting appropriate ranges for the variables. The quadrant elevation and initial position must be varied for each round and the air pressure and the air temperature must be varied for every burst (every ten rounds). The ranges for each of these variables are determined by examining historical conditions and variable values during test firings at YTC. The function “jonF” takes these ranges as input and for each script, randomly generates these values according to appropriate Uniform distributions. The ranges for these values can be seen in Table 3.2.

	Experiment A		
Variable name	Change per round/burst	Lower Limit	Upper limit
QuadrantElevation (Degrees)	round	-4.98	-4.557
X Position (meters)	round	-6	6
Y Position (meters)	round	-1006	-994
Z Position (meters)	round	94	106
Air Pressure (milibars)	burst	980	1020
Air Temperature (Celsius)	burst	15.5	32.22

Table 3.2. Table of variable ranges for Experiment A.

Using the ranges shown in Table 3.2 and the process described in Section A of this chapter, the data required for this thesis are generated.

After generating the data, differences in the virtual target impacts between ASCORE and PRODAS are compared and the round matching data interpreted. For rounds which ASCORE is able to match, an initial fire time and impact time and location are recorded, as are the distance between the two-dimensional ASCORE impact location on the VT plane and the VT PRODAS impact location. In addition, for each round a binary variable is recorded with 1 indicating ASCORE matched initial fire time with an impact and 0 otherwise. With these two additional variables a data set, as shown in Table 3.3, is created containing all of the variables and information needed to conduct a preliminary statistical analysis of the simulated test firing.

File	Air Press	Air Temp	QE	INIT.X	INIT.Y	INIT.Z	Target	Virtual Target Distance
FILE0001	995.9	30.32	4.56	3	-1005	104	0	0.0348382
FILE0002	995.9	30.32	4.73	0	-996	97	1	0.0675906

Table 3.3. Example of data file used for initial analysis

Table 3.4 shows the first two rows of the data frame used for preliminary statistical analysis. Each row in this data frame represents a simulated round and each column shows the value for each variable associated with that round. The variables in Table 3.4 are defined as follows:

- AirPress (AirPress is the variable representing the air pressure; this variable stays constant for each burst)
- AirTemp (AirTemp is the variable representing air temperature; this variable stays constant for each burst)
- QE (QE is the variable representing the quadrant elevation of the gun during the firing of a particular round. This variable is varied every round)
- INIT.X (INIT.X is the initial X position (or meters left/right) of the helicopter relative to the center of the target. This value is varied every round.)
- INIT.Y (INIT.Y is the initial Y position (or distance down range) of the helicopter relative to the center of the target. This value is varied every round.)
- INIT.Z (INIT.Z is the initial Z position (or altitude) of the helicopter relative to the center of the target. This value is varied every round)
- HitTarget (HitTarget is the variable that tells if a round impacted the target. A value of 1 represents a hit on the target, 0 for otherwise.)
- Distance (Distance is the calculated distance between the PRODAS and ASCORE virtual target coordinates.)

C. PRELIMINARY ANALYSIS

A preliminary analysis of the data is perfromed to determine if any new variables should be constructed or if any new data sets should be generated before proceeding to the final analysis. Based on 500 rounds and some exploratory data analysis, several interesting plots merit further investigation. These plots, shown in Figures 3.2 and 3.3, are the box plots of Distance for HitTarget=T and for HitTarget=F and the box-plots of INIT.X for HitTarget=T and for Hit=F respectively.

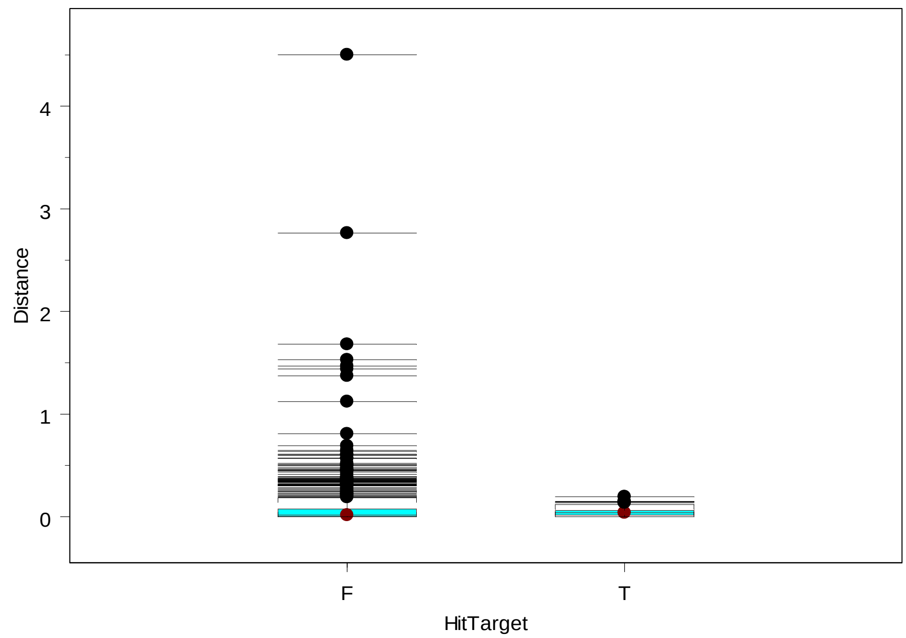


Figure 3.2. Box-plot of Distance versus HitTarget

The box plot in Figure 3.3 shows that the variance in the distance in the rounds that impact the target is much less than the distance of the rounds that miss the target. In fact, no round that impacts the target has a distance greater than .5 meters, while rounds that do not impact the target routinely exceed .5 meters and are as large as high as 5 meters.

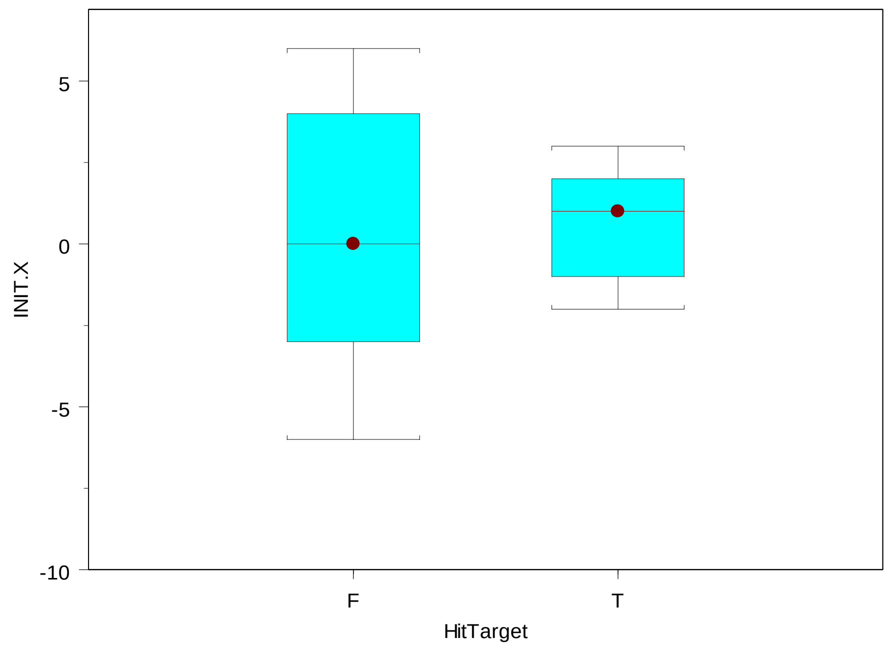


Figure 3.3. Box-plot of INIT.X versus HitTarget.

The box plot in Figure 3.6 shows that the variance of INIT.X for the rounds that impact the target is much smaller than that of the rounds that impact the target. In fact, no round with an initial INIT.X value greater than 1.5 meters than or less than -1.5 meters hit the target.

To better evaluate the results of this experiment, 20 additional bursts are generated at two new helicopter positions (Experiments B and C). Ten bursts are generated at each new position. The first new position of the helicopter is at (0,-1000, 60) meters from the target (Experiment B). This position is 40 meters below the original helicopter position of (0,-1000, 100). At this new position the approximation that ASCORE uses to score hits on the virtual target will not change. However, the quadrant elevation, target range, and INIT.Z will change. The second new position is at (0,-750,100) meters from the target (Experiment C). This position is at the same altitude and is 250 meters closer to the target than the original helicopter position. This new position is selected because the approximation that ASCORE uses to calculate a hit is different if the helicopter range is closer than 750 meters. With the helicopter hovering at 750 meters, varying the value of INIT.Y changes the range of target for each round simulated in Experiment C to values both above and below 750 meters. This will allow an evaluation of the performance of the two approximations ASCORE uses to calculate ballistic trajectories work. The range of values used in the scripting function and jonF when generating the trajectories for experiments B an C have the same units as in Experiment A and are shown in Table 3.7.

Experiment B		
Variable Name	Lower Limit	Upper Limit
Quadrant Elevation	-2.71	-2.33
X Position	-6	6
Y Position	-1006	-994
Z Position	54	66
Air Pressure	980	1020
Air Temperature	15.55	32.22
Experiment C		
Variable Name	Lower Limit	Upper Limit
Quadrant Elevation	-7.28	-6.54
X Position	-6	6
Y Position	-751	-739
Z Position	94	106
Air Pressure	980	1020
Air Temperature	15.55	32.22

Table 3.4. Tables of variable ranges for Experiments B and C.

With a cursory examination of Experiment A and the data obtained from Experiments B and C, several new predictor variables suggest themselves. These new variables are:

- TgtRang750 (This is a binary variable that takes on value 0 if the range from the target is less than 750 meters and 1 if it is greater then 750 meters)
- Exp (This variable is categorical variable that takes on value A, B, or C depending on which experiment a round belongs to)

- HitPVT (This is a binary variable that takes on a value of 1 when the round in question hits the PRODAS virtual target in the 10x10 square foot area that represents the real target)
- HitAVT (This is a binary variable that takes on a value of 1 when the round in question hits the ASCORE virtual target in the 10x10 square foot area that represents the real target)

In the next chapter, we investigate the accuracy of ASCORE bases on observations from all three experiments and using the variables described in Section C of this chapter.

IV. ANALYSIS

A. UNMATCHED/MATCHED

1. Exploratory

We first examine the factors that affect ASCORE's matching of rounds. For this part of the analysis all 700 generated rounds are used. The data is used to generate a series of plots that show the relationship between all of the variables. Only the plot that shows altitude (INIT.Z) versus Match has any interesting characteristics. This plot can be seen in Figure 4.1, where some randomness is added to the vertical axis (i.e. the vertical axis values are jittered), allowing a better feel for the proportion of rounds that are unmatched within each experiment. This plot shows the disparity in the number of rounds that are unmatched as a function of helicopter altitude.

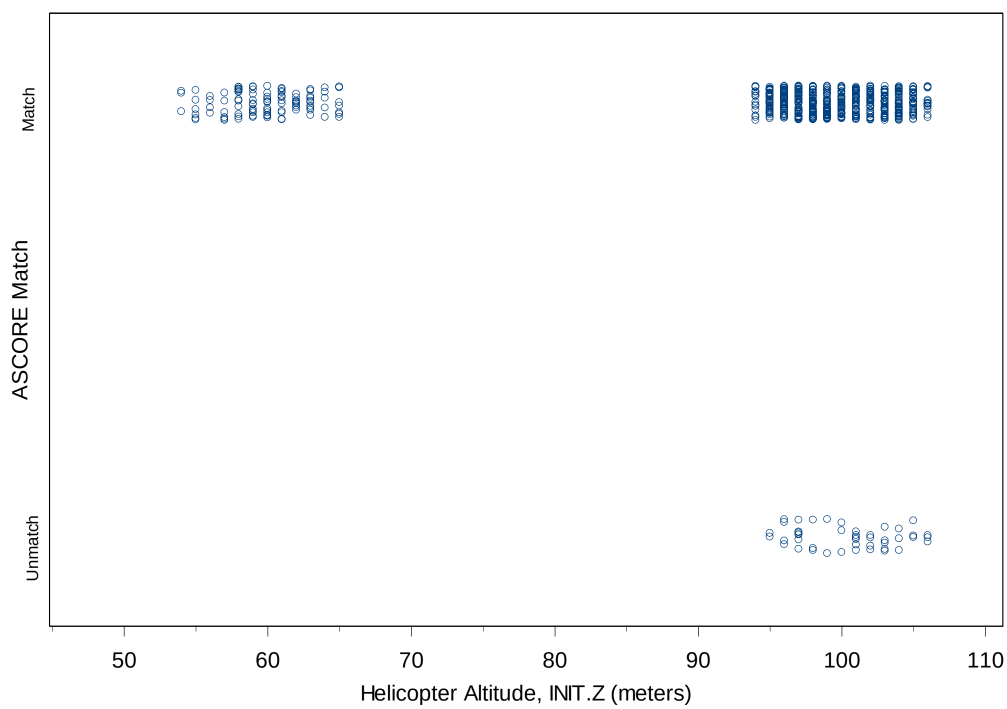


Figure 4.1. Helicopter Altitude vs MATCH

We note that Experiments A and C rounds are fired with INIT.Z greater than 90 meters and that Experiment B rounds are fired with INIT.Z less than 70 meters. Since round matching occurs within a burst, it seems that an appropriate experimental unit is a burst rather than a round. The proportion of bursts with unmatched rounds for each experiment can be seen in Table 4.1.

Experiment	Number of Bursts	Proportion of Bursts With Unmatched Rounds
A	50	0.80
B	10	0.00
C	10	0.40

Table 4.1. The proportion of unmatched rounds for each experiment.

From Table 4.1, it is unclear whether the difference in the proportions of unmatched bursts is due to a systematic difference in those experiments or by chance. To help answer this question Fisher's Exact Test for the null hypothesis of homogeneity of the proportions of unmatched bursts in three experiments is used [Ref.7]. This gives a p-value of .1882, indicating a lack of evidence that there is a systematic difference in the proportion of matched burst in the three experiments.

For completeness, a logistic regression model [Ref. 8] is also fit to this data. Here the response variable is the binary variable Match, which is 1 if a round is matched and 0 otherwise. The explanatory variables used in the logistic regression model include those listed in Chapter III, Table 3.2 along with some interaction terms. In particular, initial exploration of the data seems to indicate that the relationship between the proportion of hits and INIT.Z and QE might be different for Experiments A, B, and C. Therefore interaction between initial location and QE with the categorical variable Exp is likely. The logistic regression model "links" the probability of match p to the k explanatory variables $x_1,..,x_k$ through the logistic link function as follows:

$$\log(\frac{p}{1-p}) = \beta_0 + \beta_1 x_1 + .. + \beta_k x_k , \tag{4.1}$$

where $\beta_0,..\beta_k$ are the unknown parameters.

To eliminate unimportant and redundant explanatory variables and interactions a stepwise regression is used. The criteria for adding or removing variables at each step is

Akaike's Information Criterion (AIC), which is a likelihood criterion penalized for the number of predictor variables [Ref. 9]. The final model selection made by AIC stepwise regression produces the results shown in Table 4.2.

Coefficients	Value	Std Error	z-value
Intercept	6.499	7.624	-0.852
QE	-3.326	1.257	-2.646
INIT.Z	-0.075	0.051	-1.483
TgtRang750	0.685	1.024	0.669
HitTarget	0.231	0.453	-0.511
ExpB	12.461	11.961	1.041
ExpC	-5.229	2.694	-1.941
HitTarget:ExpB	-0.023	22.607	-0.001
HitTarget:ExpC	-2.348	1.100	-2.134
Null Deviance: 334.04 on 699 degrees of freedom Residual Deviance: 304.77 on 691 degrees of freedom			

Table 4.2. Resulting model of AIC stepwise regression.

For this model we use the likelihood ratio test to test the null model that $\log(\frac{p}{1-p})$ is constant. The likelihood ratio test statistic is found by calculating the difference in the null and residual deviance which is 29.27. Under the null hypothesis, this test statistic has an approximate Chi-squared distribution with 8 degrees of freedom, which gives a p-value of .002114. This result is strong evidence against the null hypothesis. Therefore there is at least one important predictor included in Table 4.2.

We suspect that some or all of these variables change the probability of round matching. We also suspect that a major factor contributing to the errors in round matching process seen in these data is due to a programming error in the version of ASCORE currently in the possession of NPS (Barbara Carlson, Personnel Communication, May 20, 2005). This error has been corrected in the version of ASCORE used by YTC. Thus, a true evaluation of the ASCORE's matching algorithm cannot be conducted without first acquiring a copy of YTC's corrected version of ASCORE.

B. MATCHED DISTANCE

The goal of this part of the experimental analysis is to try and determine which variables have the greatest effect on the distance between the “true” virtual target impact location of PRODAS in the corresponding approximate virtual target impact locations computed by ASCORE for the round. For this part of the analysis, all of the rounds that are unmatched by ASCORE are removed from the data set, leaving 655 matched rounds (out of 700) in Experiments A, B, and C.

1. Exploratory Analysis on PRODAS VT and ASCORE VT Impacts

The first step in this part of the analysis is to determine which rounds generated by PRODAS impact the virtual target but miss the real target. Identifying these rounds is important because all rounds that hit a defined 10ft x 10ft area on the virtual target can be scored as a live fire hit for the purpose of weapons validation even though they may have missed the real target. Identifying rounds that impact the virtual target and miss the real target is the purpose for generating virtual target impacts. Table 4.3 shows that 27 rounds fall into this category. In addition, because the 10ft x 10ft virtual target is perpendicular to the line of fire, no round that impacts the real target should miss the virtual target. Table 4.3 shows that this is the case for all rounds.

	Miss Target	Hit Target
Miss PRODAS VT	506	0
Hit PRODAS VT	27	122

Table 4.3. Matrix of PRODAS virtual target round impacts.

The next step in analyzing PRODAS and ASCORE outputs is to see if there are any major discrepancies between the PRODAS virtual target impacts and the ASCORE virtual target impacts. An analysis of the data shows that every trajectory generated by PRODAS whose calculated virtual target impact occurred in a predefined 10ft x 10ft area in the virtual plane, was also calculated by ASCORE to be in the same predefined area of the virtual plane. Additionally, ASCORE did not score any rounds in the predefined virtual target area that were not computed to be there by PRODAS. This shows that ASCORE’s calculation of the virtual target impact location, for the data generated in this thesis, is close enough to the “true” virtual target location that no scoring errors occurred.

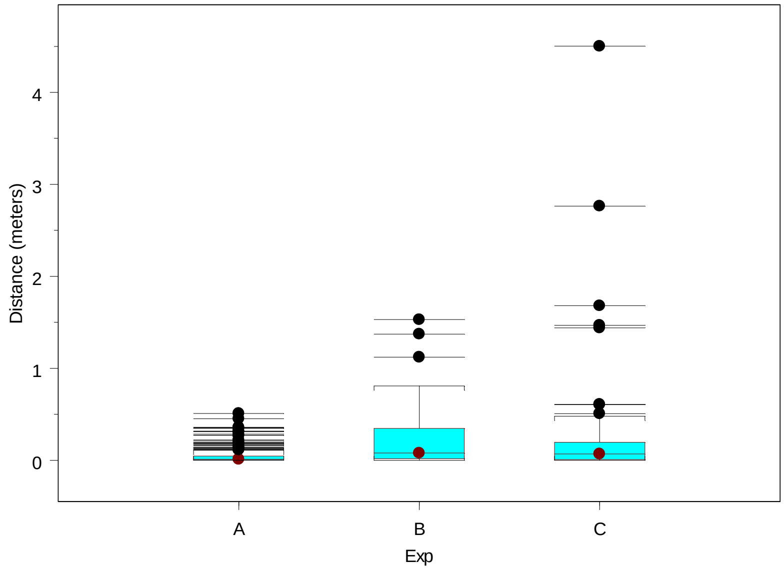


Figure 4.2. Distance for Experiments A, B, and C.

We next examine the distance (Distance) between the PRODAS virtual target impact location and the corresponding approximate virtual target impact location computed by ASCORE. Figure 4.2 gives the box-plots of Distance for each of the three experiments. The distributions of Distance are right skewed with much greater variability for Experiments B and C (each of which only has 100 observations). In particular, Experiment C has great deal of variability. This is of particular interest because it is suspected that there is some inconsistency in the two approximations ASCORE uses to calculate virtual target impacts. To further investigate the cause of increased variability in Experiment C, the box-plots in Figure 4.3 show the distribution of Distance for range to target (TgtRang750) below and above 750 meters.

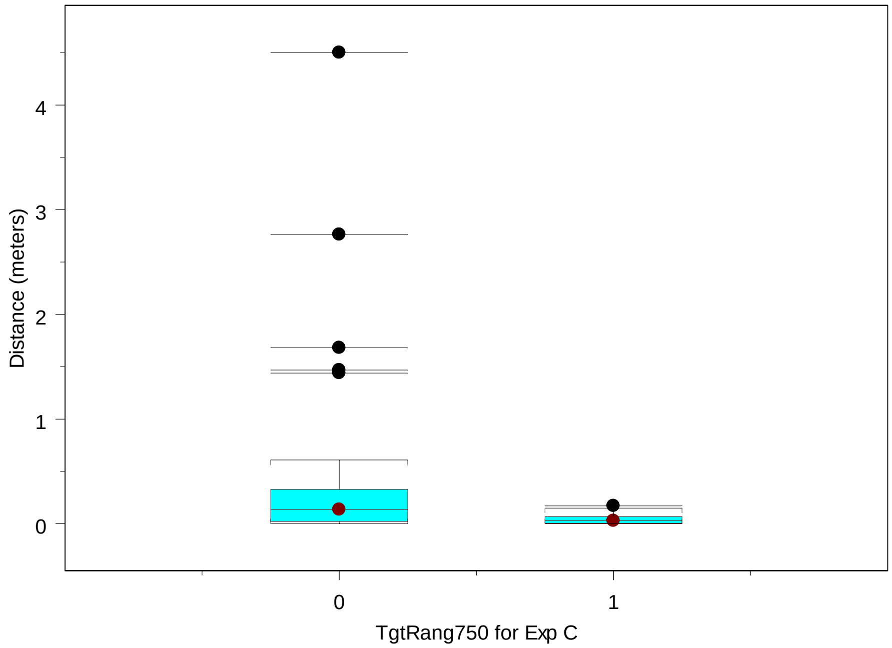


Figure 4.3. Distance as a function of target range above (TgtRang750=0) and below (TgtRang750=0) 750 meters.

Figure 4.3 suggest that Distance is greater and has more variability for rounds with target range less than 750 meters than those above 750 meters.

Exploration of the data suggests that there is a possible link between the distance that a round impacts from the target and Distance. To capture the effect of this relationship in the analysis, a new predictor variable called Dist.Impact is created whose value is the distance from the center of the target to the ground impact location of all rounds that miss the real target.

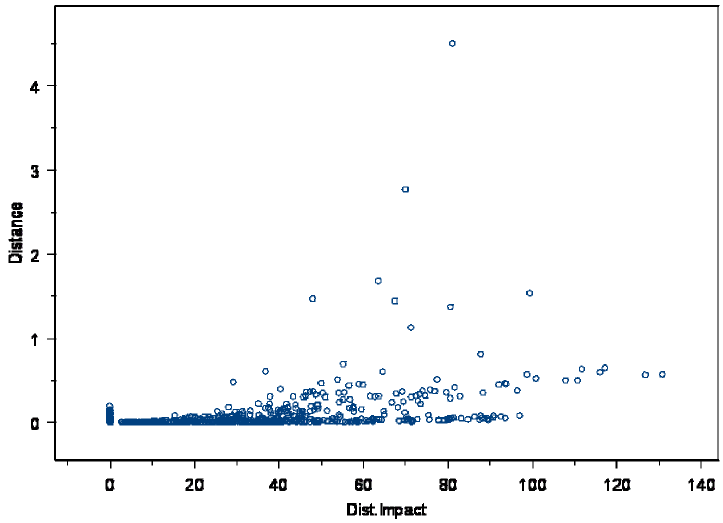


Figure 4.4. Response variable Distance as a function of Dist.Impact.

Figure 4.4 shows the response variable Distance plotted against the predictor variable Dist.Impact. Figure 4.4 shows that there is an increasing relationship between the distance that a round misses the real target (Dist.Impact) and the difference in the ASCORE and PRODAS virtual target calculations Distance. However, how Distance increases with Dist.Impact is unclear. Figure 4.5 helps to explain this by showing the relationship between Distance and Dist.Impact for each experiment.

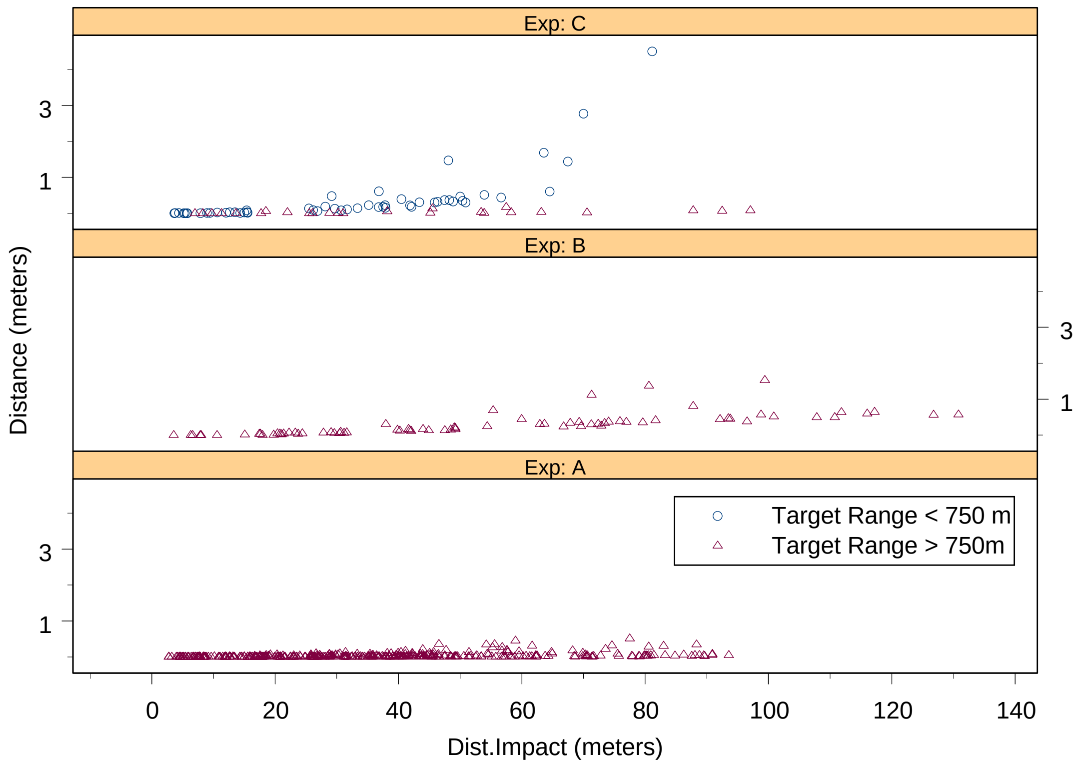


Figure 4.5. Distance versus Dist.Impact for Experiments A, B, and C.

Figure 4.5 confirms that Experiment C contains more variability than the other experiments. Additionally, in can also be seen that Distance increases much more rapidly with Dist.Impact in Experiment C when target range is less than 750 as compared to the other experiments.

To test the null hypothesis that the mean of the distance in the rounds above and below 750 meters is equal for Experiment C, the Wilcoxon Rank Sum Test is used [Ref. 7]. Asymptotically, when there is no difference between groups the Wilcoxon Rank Sum Test statistic has an approximate standard normal distribution. The computed test statistic is -3.96 with a corresponding p-value of .0001. This is strong evidence that there is a difference in the mean distance of the rounds above and below 750 meters, and supports the claim that the algorithm used by ASCORE to calculate virtual target impacts when the target range is less than 750 meters has a marked effect on ASCORE accuracy.

2. Model

With the exploratory analysis complete, a linear regression model is again fit using only the 655 matched rounds. The first model fit is a simple linear regression model with Distance as the response variable. The predictor variables are all variables listed in Table 4.2, the new variable Dist.Impact, and some interaction terms. The plot of the residuals versus the fitted values for this model, seen in Figure 4.6, shows signs of severe heteroscadcicity suggesting the need for a transformation of the response variable to stabilize the variance.

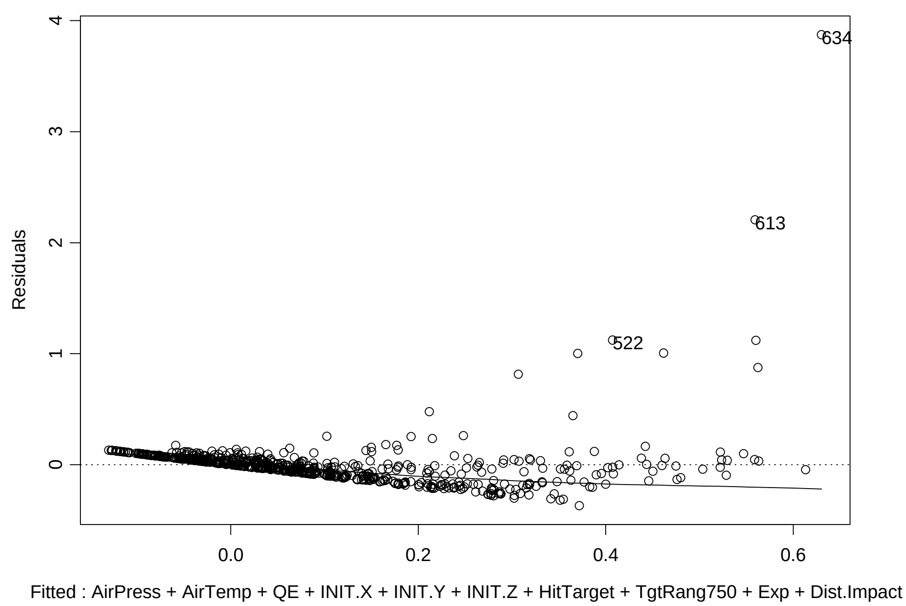


Figure 4.6. Residual plot of fitted model.

Experimentation makes it is clear that with a square root transformation the variance of the residuals still contains a significant amount of heteroscadcicity. However, the log transformation shows some small decrease in the variability of the residuals with the fitted values. Careful choice of a power transformation, such as using a Box-Cox transformation [Ref. 8] to estimate the power needed to stabilize the variance, would yield a transformation much closer to a log transformation than a square root transformation. Because both ASCORE and PRODAS computations are deterministic and based on the physics of trajectories, it is strongly suspected that the relationship between distance and the explanatory variable is not intrinsically linear. Thus, it is likely that no simple transformation of the response variables Distance will yield a model which fits the data as well as we would like. However, for the purposes of this thesis using a log transformation for Distance adequately stabilizes the variance, and with this transformation we can see which of the predictor variables are most important.

Partial residual plots for the numeric predictor variables indicate that there are slight nonlinear relationships between expected log Distance and the partial effect of the numeric predictors QE, INIT.Z, and Dist.Impact. By using AIC stepwise regression and starting with a model with interactions and quadratic terms for QE, INIT.Z, Dist.Impact, INIT.X, and INIT.Y gives a model whose estimated coefficients are in Table 4.4.

Coefficients:	Value	Std.Error
(Intercept)	-11.9501	2.3234
I(Dist.Impact^2)	-0.0004	0.0001
INIT.X	0.0256	0.0151
TgtRang750	-0.8883	0.4457
I(QE^2)	0.0397	0.0146
HitTarget	3.7679	0.2148
AirTemp	0.0631	0.0107
INIT.Y	-0.0032	0.0022
Dist.Impact:TgtRang750	-0.0373	0.0122
Dist.Impact:INIT.Z	0.0012	0.0002
Dist.ImpactExpB	0.1537	0.013
Dist.ImpactExpC	0.0246	0.0213
I(Dist.Impact^2)ExpB	-0.0008	0.0001
I(Dist.Impact^2)ExpC	-0.0003	0.0002
Residual standard Error	1.344 on 641 df	
Multiple R-Squared	0.6228	
F-statistic:	81.43 on 13 and 641 df	
p-value	0	

Table 4.4. Resulting model of AIC stepwise regression.

3. Discussion

The final model, shown in Table 4.4, contains most of the originally included predictors. The only variable that is eliminated from the model completely is air pressure.

Because the experiment was designed so that air pressure is linearly independent of all other predictors, we can conclude that air pressure is the only variable in the model that has the same impact in both ASCORE and PRODAS computations. All other predictors in the model have some effect on the difference in the distance between the ASCORE and PRODAS virtual target calculations.

We note that Exp appears in the final model. Regressions were fit separately for each experiment. However, these yielded results consistent with the regression fit for the pooled data. In particular, log transformations for Distance were used to stabilize the variance of the response in models for each of the three experiments. Subsets of the same predictors in Table 4.4 appear in the regression equations for each of the experiments.

The effect of Dist.Impact seems to be the most interesting variable in the regression. To further explore the relationship between Dist.Impact and the response for Experiment A, the estimated expected log Distance along with 95% confidence limits are plotted in Figure 4.7 holding all of the other predictor variables constant (Table 4.5) while varying Dist.Impact.

AirTemp	23.541
QE	-4.78
INIT.X	0.00
INIT.Y	-1000
INIT.Z	100
TgtRage750	1 (>750)
Exp	A
HitTarget	0 (Miss)
Dist.Impact	Vary

Table 4.5. Table of variable values used to construct confidence interval.

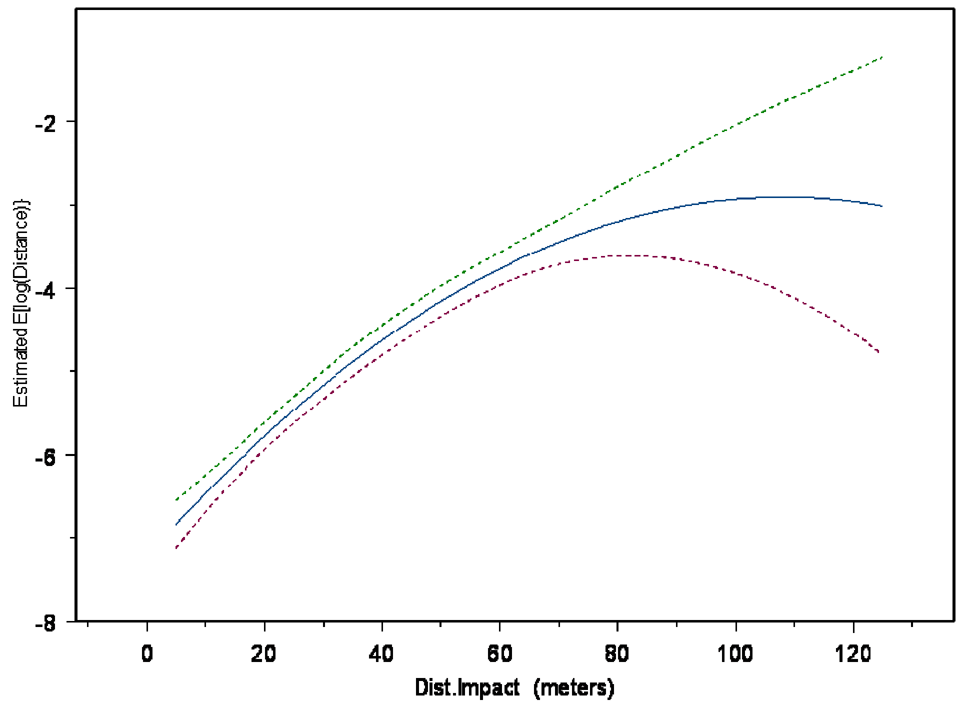


Figure 4.7. Confidence interval for Expected Log Distance as a function of Dist.Impact.

The point-wise confidence intervals, graphed in Figure 4.7, confirm the suspicion that the expected log Distance increases as a round impacts further from the target. Also, because most of these simulated rounds have Dist.Impact less than 100 meters, it is not surprising that there is much less certainty in the estimated expected log Distance for larger values of Dist.Impact. In fact, it is doubtful that expected log Distance decreases for large values of Dist.Impact (as depicted in Figure 4.7). We note that the fitted values based on the estimated coefficients of Table 4.4 give estimates of expected log Distance rather than the expected Distance. With the model used in Table 4.4, Distance has a log-normal distribution with parameters $\mu = E[log(Distance)]$ and $\sigma^2 = Var[log(Distance)]$ [Ref.7]. The expected value of Distance is $E[Distance] = \exp\{\mu + \sigma^2/2\}$. Thus we can obtain the Maximum Likelihood Estimate (MLE) of expected log Distance by substituting the fitted values from the regression of Table 4.4 for μ and a residual standard error of 1.344 from Table 4.4 for σ. Figure 4.8 plots the MLE of expected Distance as a function of Dist.Impact for all Experiments.

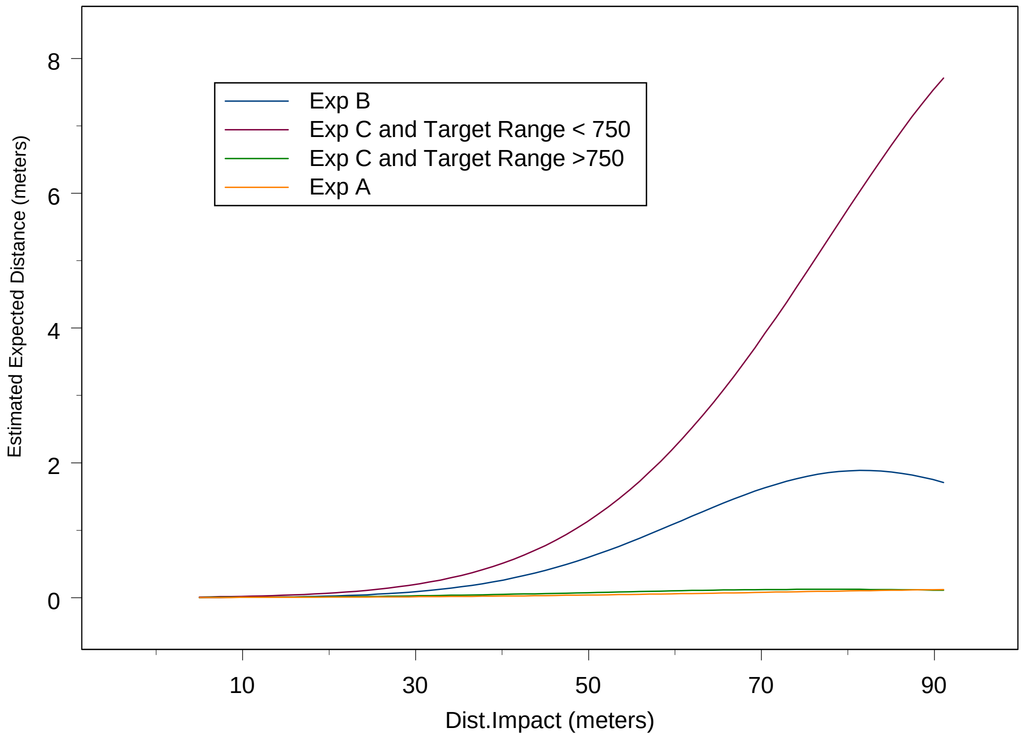


Figure 4.8. Estimated Distance as a function of Dist.Impact for all Experiments.

Figure 4.8 shows that the expected Distance of a round increases as the impact location of a round moves further from the target. However, the rate at which the expected Distance changes differs significantly between experiments. The change is particularly large in expected Distance for Experiment C for rounds that have a target range less than 750 meters. This supports the idea that the accuracy of the ballistic calculation ASCORE uses to compute the trajectories of scored rounds is worse for rounds with a target range less than 750 meters.

A approximate 95% confidence interval (shown in Figure 4.9), for Experiment A, can be constructed for expected Distance by combining the 97.5% confidence intervals for expected log Distance with the 97.5% confidence interval for variance of the expected log Distance and using Bonforinni's inequality.

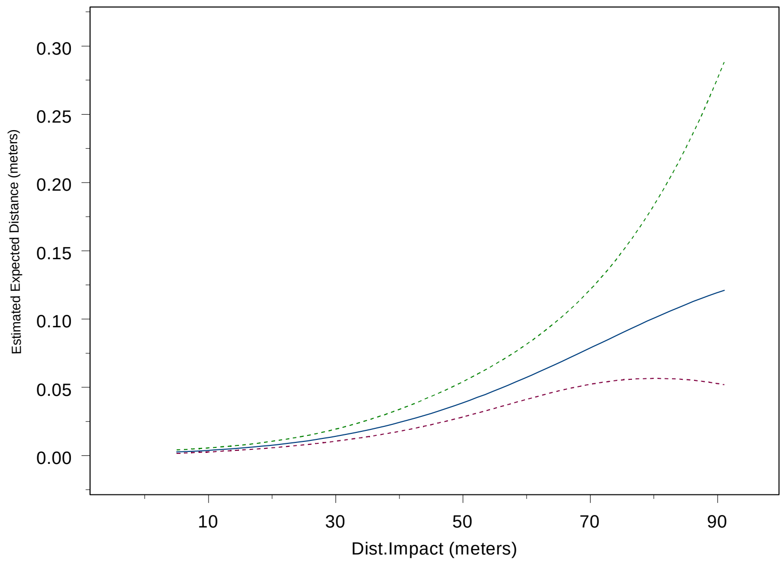


Figure 4.9. Experiment A 95% confidence interval for E[Distance].

The increase in Distance as Dist.Impact increases can still be seen in Figure 4.9. Additionally, the confidence interval widnes for larger values of Dist.Impact. This is due to the fact there are fewer observations with large values for Dist.Impact. Confidence intervals computed for Experiments B and C were computed using similar methods, and have similar shapes.

It is important to consider that the data generated for this thesis were tightly controlled in that the majority of the rounds hit within 80 meters of the target. All conditions and variable settings used in generating the trajectories were selected in a manner that would be simple and ideal for processing in ASCORE. Another factor to consider is that in this study the actual "truth" about the path and impact of the trajectories as generated by PRODAS is given to ASCORE. In a real test firing, not only would there be a considerable amount of error introduced in gathering the data as mentioned in Chapter I, but the rounds can have a miss distances substantially larger than 80 meters. Thus, the study here represents a "best-case scenario." Constructing a new series of trajectories with larger Dist.Impact values could be generated and used to better evaluate the effect of large Dist.Impact values on expected Distance and should be done in future research.

V. CONCLUSIONS AND RECOMMENDATIONS

This thesis focused on using PRODAS to evaluate ASCORE. This was accomplished by selecting and varying environmental and ballistic variables common to both ASCORE and PRODAS. Round trajectories were generated in PRODAS along with the “true” virtual target impact location. The simulated rounds were treated as if they had been observed from live fire tests and ASCORE was used to match rounds of 10 round simulated bursts and approximate their virtual target impacts.

ASCORE failed to match 45 of the 700 rounds simulated in PRODAS. It appears that the proportion of unmatched rounds varies depending primarily on QE and INIT.Z. The NPS version of ASCORE has several formatting errors that are the most likely the cause of mismatched rounds in this experiment. These formatting errors are corrected in YTC’s version of ASCORE. However, mismatched rounds are still a concern to YTC, and this problem needs to be addressed. For 655 out of 700 rounds which were matched by ASCORE, we first noted that PRODAS generated 27 trajectories which missed the real target but hit the 10ft x 10ft virtual target. When evaluated by ASCORE, each of these 27 rounds were scored as having hit the 10ft x 10ft virtual target. Similarly, all of the rounds that missed the virtual target in PRODAS were scored by ASCORE as having missed the 10ft x 10ft virtual target. Thus, for these experimental conditions, ASCORE evaluated the weapons system with 100% accuracy.

In general, ASCORE’s ballistic calculations of the virtual target impact location were close to those of PRODAS. These experiments were simulated under controlled settings, whereas live fire tests are considerably more complex. The analysis in this thesis suggests that changes that increase the complexity of the test firing, will result in a decrease in ASCORE’s ability to correctly calculate the virtual target. The error in ASCORE’s virtual target impact location increased with the distance that rounds miss the target. This relationship was strongest when the helicopter range from the target was less than 750 meters. At these ranges ASCORE uses a different algorithm. The ASCORE error with target miss distance is of concern because the live fire miss distances are typically much greater than those simulated here. The Navy is considering using ASCORE to aid in fire control. Before ASCORE can be confidently used by the Navy for fire control purposes, it must be more carefully evaluated over a wider range of conditions to more fully evaluate the error in the virtual target calculations.

A. FUTURE RESEARCH

This thesis was just the first step in this area of research. Much of the effort in this work was concerned with acquiring and writing necessary programs, developing methods to integrate these programs, and documenting them appropriately. This thesis, and the documentation found in the appendixes serve as a spring board for future research.

1. General Areas of Difficulty

There are two major issues that still need to be resolved before full scale experimentation of ASCORE using PRODAS can proceed:

- Communication with Arrow Tech (The makers of PRODAS).

The ability to automate as much of the data generation process as possible is a critical component in constructing a data set large enough to conduct meaningful analysis. Automating trajectory generation within PRODAS, beyond that which is documented in Appendix B, will require the help of Arrow Tech and the addition of new modules to PRODAS. Neither YTC or NPS have been able to ellicit the required support from Arrow Tech. Due to this lack of support we recommend that a different ballistic package be used to evaluate ASCORE.

- Updating the NPS version of ASCORE.

The version of ASCORE currently in the possession of NPS lacks some of the additional modules that format the input files. Without these programs, almost any attempt to input data by hand will be misread in some form by ASCORE. Since there is no documentation for the ASCORE input files the only way to resolve this issue is by consulting with YTC. Before future use of ASCORE takes place, the full set of updates for ASCORE should be loaded onto the NPS version.

2. Unresolved Programming Issues

Three specific problems need to be resolved to better evaluate ASCORE and use PRODAS. They are:

- If PRODAS is used in the future then it will be necessary to coordinate with Arrow Tech to automate the running of script files. The current NPS version of PRODAS requires the user to manually produce round trajectories one at a time. This process can be automated by the addition of a new module PRODAS. The ability to do this would not only save a tremendous amount of time, but it would also allow the generation of much more data.

- Fix the MET module

The meteorological tower input files in ASCORE have a formatting problem that needs to be resolved. Fixing this problem will result in the ability to vary wind speed and wind direction when processing data in ASCORE.

- Round matching inputs.

During the course of evaluating ASCORE's round matching algorithm, it was suspected that a contributing factor to ASCORE's inability to match rounds was the result of some unknown formatting or programming error. The source of this error is suspected to be associated with synchronizing the time inputs between the rte-223.sff and flt4.tv files (please see Appendix C for a detailed description of these files).

3. Future Research Areas

1. Continued analysis of ASCORE round matching algorithm.

During the course of researching this thesis, it was discovered that ASCORE changes the velocity of a round in an attempt to force it to match a trajectory. The muzzle velocity input in ASCORE is simply used as a starting point for the round matching process. In this thesis, the muzzle velocity of a round was constrained to be no greater than 805m/s. This constraint could be effectively removed in ASCORE by modifying the ASCORE.run file (See Appendix C for more detail on this file) and allowing ASCORE to freely modify the velocity of all rounds. A more thorough study exploring the relationship between how ASOCRE matches a round and selects a corresponding velocity would be very useful.

2. More observations at multiple fire locations and angles of fire.

In this thesis, only three fire locations with the same angle of fire were used. In addition, these three locations and the experimental conditions were very favorable to ASCORE in that all rounds that missed the target missed by fewer than 70 meters. It has been shown in this thesis that the initial location of the helicopter influences the miss distance and hence the virtual target calculations for ASCORE. More experiments over a more comprehensive range of fire locations and angles is needed to provide a better overall evaluation of ASCORE for YTC.

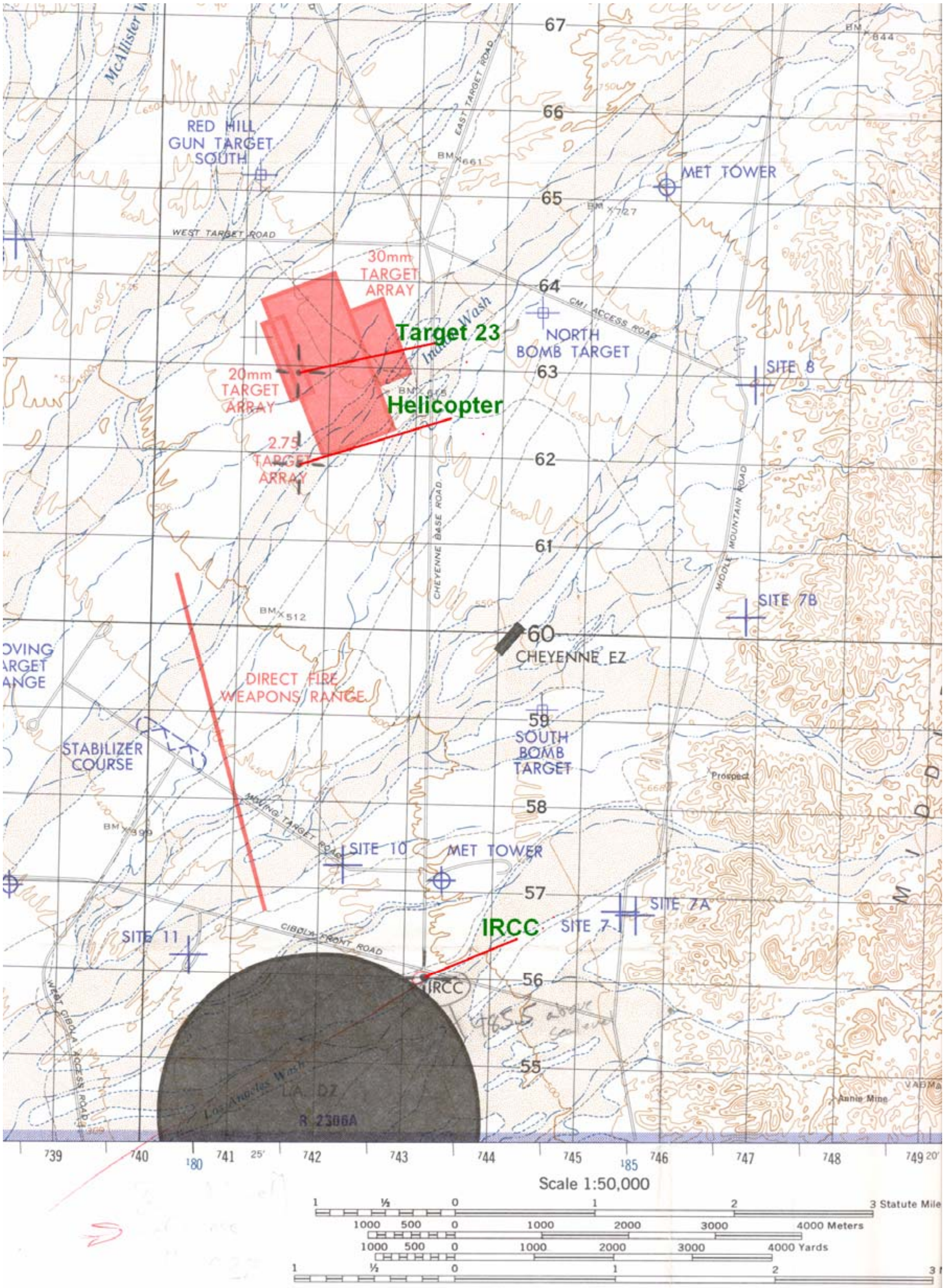
3. Virtual target calculations with ranges less than 750 meters.

In this thesis the majority of the trajectories that were generated had a target range greater than 750 meters. Some limited experimentation was done with trajectories having a range of less than 750 meters. Analysis of the data in this thesis suggest that rounds with a range to target less than 750 meters tend to have a larger error in virtual target calculations. A new study should be conducted by focusing on rounds that have target ranges less than 750 meters. This would greatly increase the current knowledge of the workings of ASCORE's ballistic algorithms work.

4. Moving Aircraft Simulations.

The experiments conducted in this thesis all deal with a helicopter that is hovering around a fixed location. Since YTC conducts a large number of test firings with moving helicopters, it would be extremely valuable to determine the effect a moving helicopter has on the ASCORE ballistic calculations.

APPENDIX A: MAP OF FIRING RANGE



THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX B: USING PRODAS

B.1 INTRODUCTION

Arrow Tech Associates Incorporated in Burlington Vermont developed Projectile Design & Analysis System (PRODAS) version 3 in April of 2002. The main purpose of PRODAS is to perform rapid evaluation on the performance of ammunition characteristics. To fulfill this purpose, PRODAS has the capabilities to both design and evaluate projectiles.

While the ability to design projectiles is a main part of the PRODAS program, we will not discuss this aspect of PRODAS because the 30 mm ammunition design is already hard-coded into the PRODAS system. Analyzing ammunition in PRODAS is what this appendix will focus on. Analyses in PRODAS include interior ballistics, launch dynamics, exterior ballistics, and terminal ballistics.

PRODAS was developed using methodologies and techniques such that predicted performance estimates are based in part on prior experimental testing. PRODAS links several diversified analysis together into a common database so that the results of one analysis feed directly to a subsequent analysis. The database is maintained such that, as experimental data becomes available, the analysis may be redone using the actual parameters instead of estimated parameters. As a simulator, PRODAS is a deterministic model.

PRODAS analysis options are diagramed in Figure B.1.

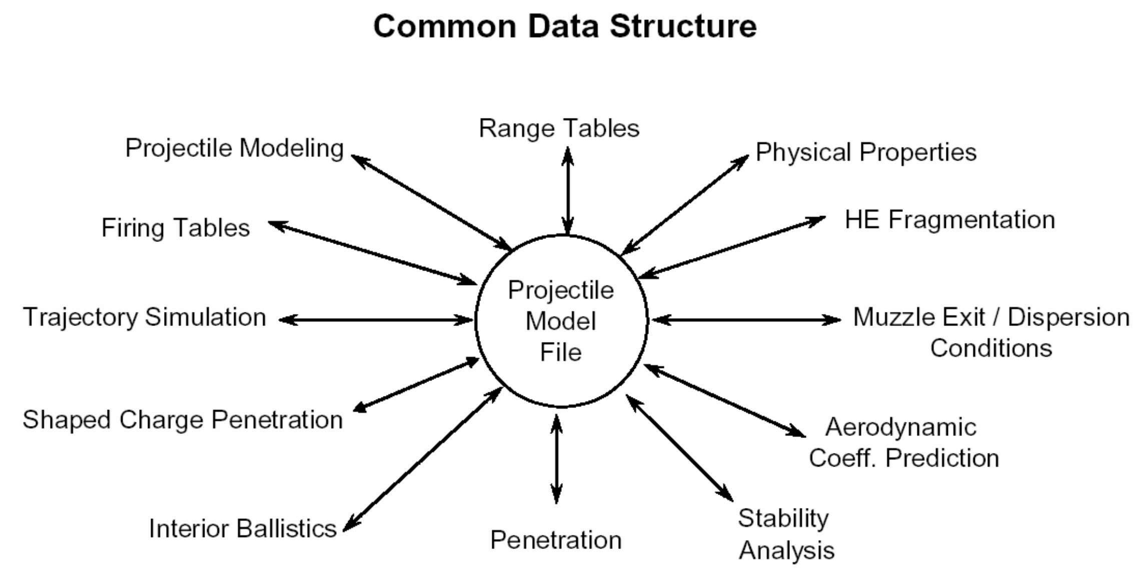


Figure B.1 PRODAS analysis options

It should be noted that not all of these analyses are pertinent to our study. This analysis data structure is a summary of all analyses and can be found in the PRODAS User Manual [Ref. 6].

This appendix is meant to assist the reader in running PRODAS with items related to this thesis. If the reader desires a more in-depth explanation of PRODAS, then the reader should refer to the PRODAS V3 User Manual [Ref. 6], the PRODAS V3 Technical Manual [Ref. 9], or the PRODAS V3 VB Script User Manual [Ref. 10]. The reader may also contact Arrow Tech through their PRODAS website at http://www.prodas.com .

B.2 LOCKED PARAMETERS vs CHANGEABLE PARAMETERS

The parameters for any projectile in PRODAS can be separated into two categories. Those that are unique to a particular ammunition type and those that are not unique to an ammunition type. The former of which will be referred to as hard coded parameters and the latter as changeable parameters. Hard coded parameters are specified for a particular round when the round is created in PRODAS.

Hard Coded Parameters:

Parameter	Description
Muzzle Velocity	The speed of a projectile as it leaves the muzzle of a weapon. Unless the projectile operates under its own power the muzzle velocity is its highest speed.

Changeable Parameters:

Parameter	Description
Airplane Velocity	The speed of the aircraft.
Airplane Altitude	The distance to the aircraft from ground.
Airplane Bank Angle	The angle between the aircraft's normal axis and the Earth's vertical plane containing the aircraft's longitudinal axis.
Airplane Dive Angle	The angle between the aircraft's flight path and the ground.
Initial X Position	The initial x position of the projectile.
Initial Y Position	The initial y position of the projectile.
Initial Z Position	The initial z position of the projectile.
Quadrant Elevation	The angle between ground and the axis of the bore when the weapon is laid.
MET Table Source	The table that reports to the common database the meteorological conditions. PRODAS has five built in conditions: Cold, Polar, Std, Tropical, and Hot. PRODAS also contains a User condition where a user can input different meteorological conditions.
Atmos Temp	Environmental temperature.
Atmos Pressure	Environmental pressure.
Wind Direction	The wind direction with 0 degrees being a tail wind.
Wind Velocity	The wind speed.
Above sea level	Distance of ground above sea level.

Figure B.2 PRODAS parameters

It is easy to distinguish between hard coded parameters and changeable parameters in PRODAS. Hard coded parameters are shaded in gray while the changeable parameters are in normal black font.

B.3 BALLISTICS IN PRODAS

The ballistics in PRODAS are based off of physics based models. Listed below are the basic projectile design considerations that are addressed in PRODAS:

A. Ammunition Type
B. Physical Constraints
   1. Geometric Properties
   2. Physical Properties
C. Exterior Ballistics
   1. Spin-Fin-Flare Stabilized

2. Aerodynamics
3. Stability
4. Time of Flight
5. Velocity
6. Accuracy
    a. Dispersion

D. Interior Ballistics
    1. Impulse
    2. Chamber Pressure
    3. Velocity
    4. Acceleration
    5. Propellant
    6. Primer
    7. Rotating Band

E. Structural Integrity

F. Terminal Ballistics

For a more complete list and detailed explanation, the reader should refer to the PRODAS V3 Technical Manual [Ref. 9].

B.4 HOW TO RUN A BASIC SIMULATION (GUI)

There is a step by step procedure that must be followed to run a PRODAS trajectory simulation for a stationary aircraft. The user should first start the PRODAS program. The next step is to open the data for the projectile the user wishes to analyze. From the toolbar, click **Open** from the **File** menu. A box similar to Figure B.3 should appear on your screen.

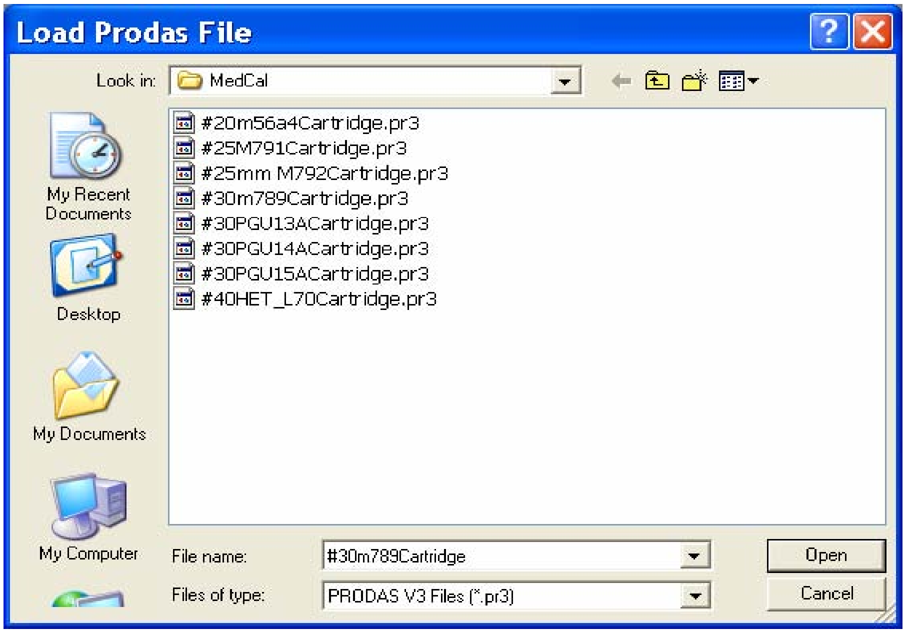


Figure B.3

Highlight the desired projectile and click **Open**. The desired projectile should then load into PRODAS and the round is ready to be analyzed. The PRODAS screen should look like Figure B.4.

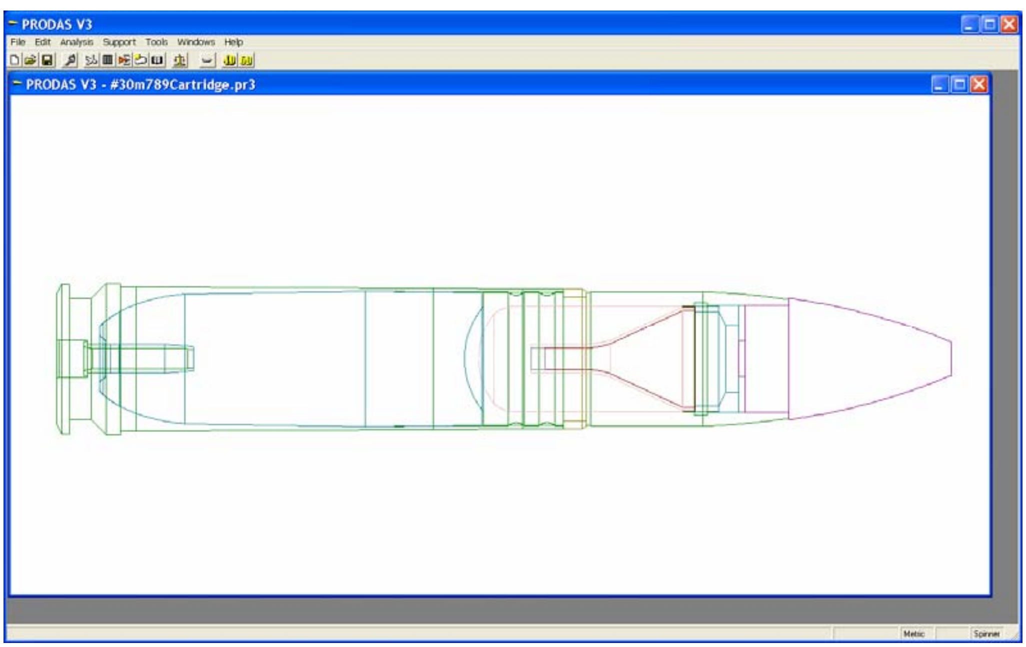


Figure B.4

Before running a trajectory simulation, the user must first perform three different analyses on the projectile. These different analyses are essential because the information from these analyses are sent to the common database where the trajectory simulation will then use information from these analyses.

The first analysis to run is the Mass Properties Analysis. Running Mass Properties will calculate the weights, inertias, and center of gravity based on the model. To open the analysis, chose **Mass Properties** from the **Analysis Menu** on the toolbar. The mass properties screen should look like Figure B.5.

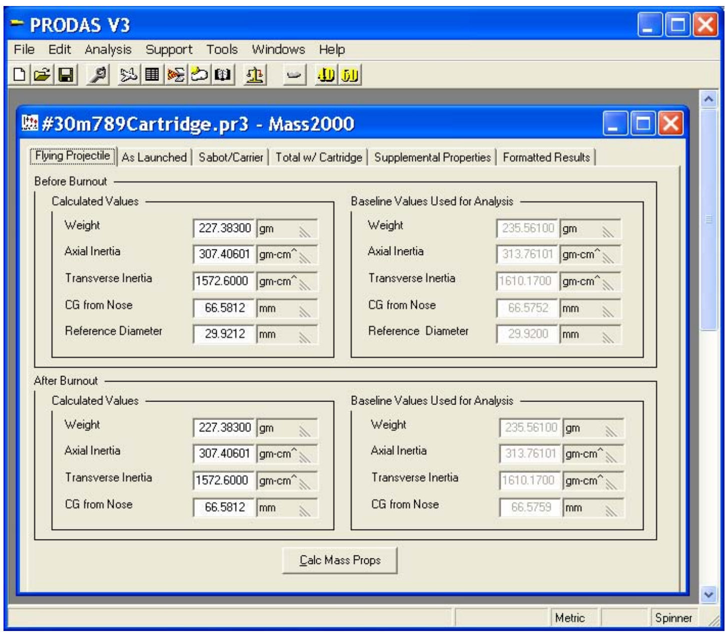


Figure B.5

To run the analysis, simply hit the **Calc Mass Props** button and minimize the mass properties box.

The next analysis to run is the Aerodynamic Predictions Analysis. The purpose of this analysis is to generate the aerodynamic coefficients. To open this analysis chose **Aerodynamic Predictions** under **Aerodynamics** in the **Analysis Menu**. Figure B.6 pictures how the aerodynamic predictions box should appear on your screen.

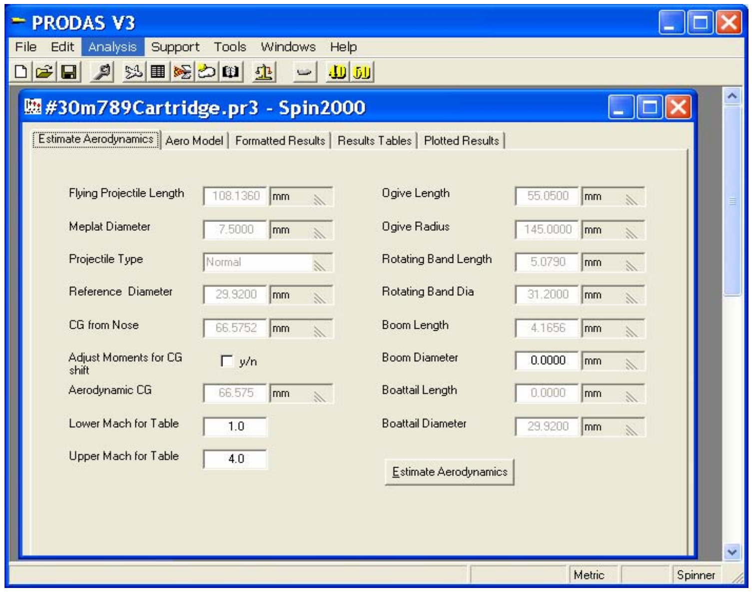


Figure B.6

To run the analysis, simply hit the **Estimate Aerodynamics** button and minimize the aerodynamics prediction box.

The next analysis to run is the Stability Evaluation Analysis. This analysis allows for the computation of gyroscopic and dynamic stability for the projectile. To open this analysis chose **Stability Evaluation** under **Aerodynamics** in the **Analysis Menu**. The stability evaluation screen is shown in Figure B.7.

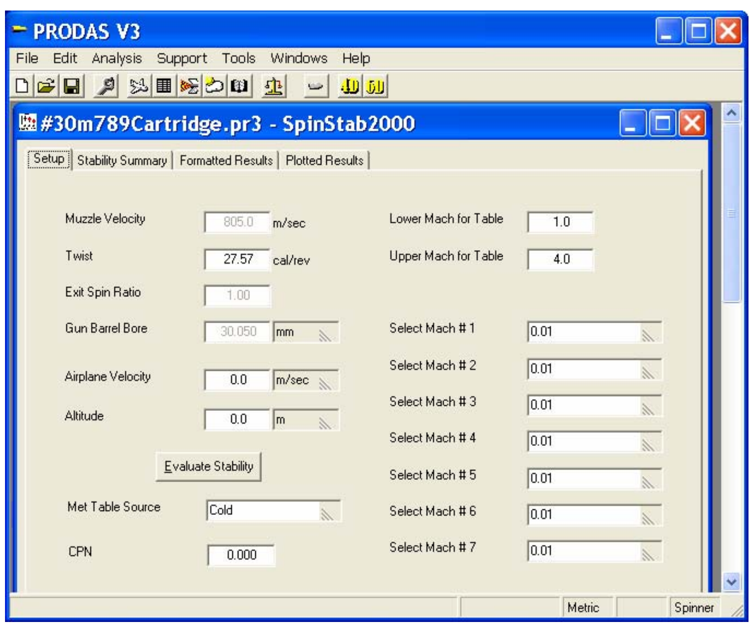


Figure B.7

To run this analysis, hit the **Evaluate Stability** button and minimize the Stability Evaluation box.

At this point, the user is ready to begin the trajectory simulation. From the **Analysis Menu** chose **Trajectories**, followed by **Fixed Plane-6D**. The box shown in Figure B.8 should appear on your screen:

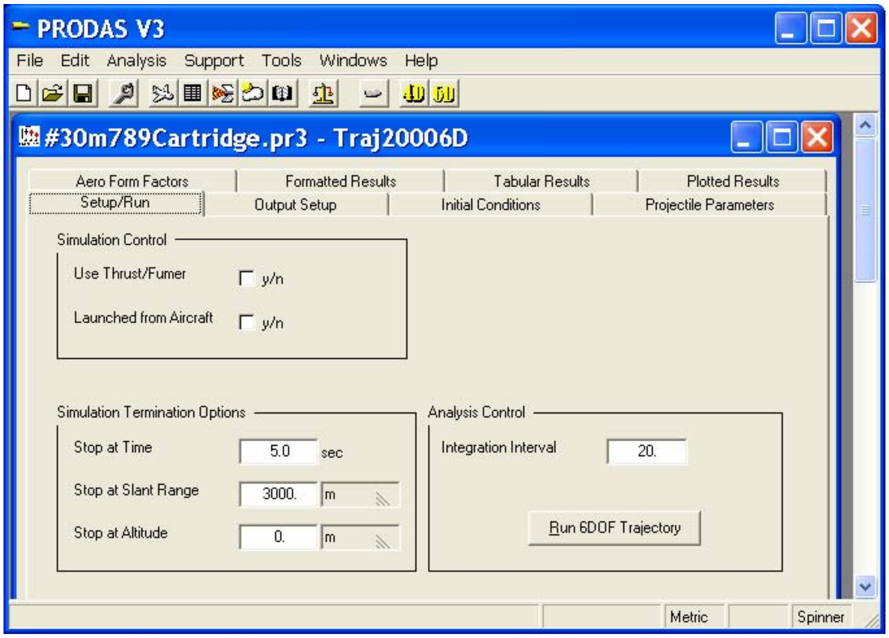


Figure B.8

Before starting the simulation, the initial conditions must be formatted. Click on the **Initial Conditions** tab and a screen similar to Figure B.9 should appear.

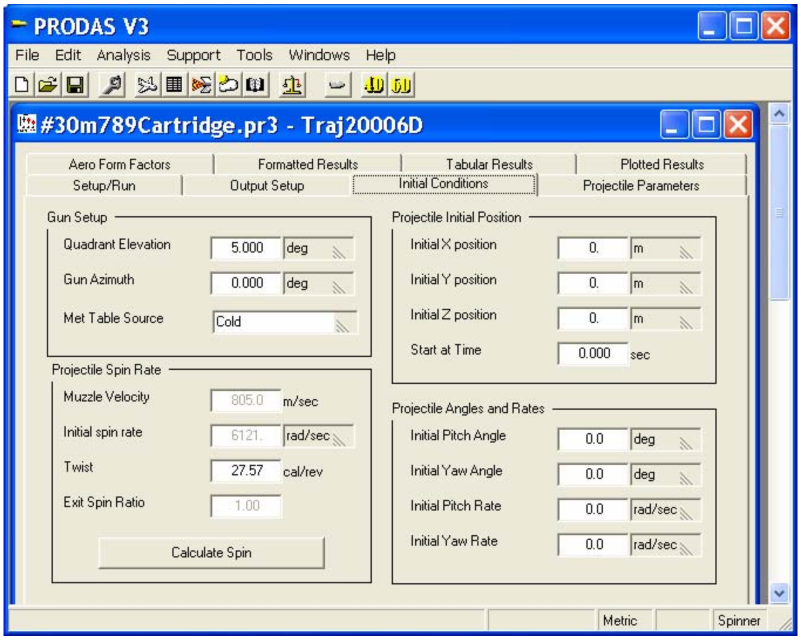


Figure B.9

Change any conditions and hit the **Calculate Spin** button. Next click on the **Setup/Run** tab. Figure B.10 depicts what the new screen should look like.

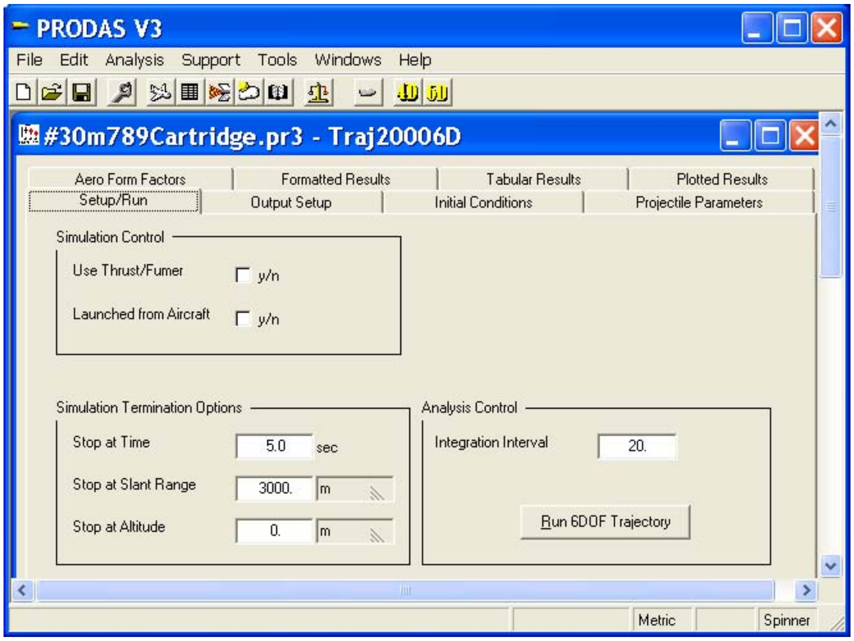


Figure B.10

If the simulation being run is with a stationary aircraft, the only items that may change are the desired Simulation Termination Options. The simulation will terminate when the first terminating condition is meet. Once these are set, run the simulation by hitting the **Run 6DOF Trajectory** button.

B.5 OUTPUTS: DIFFERENT RESULT WINDOWS

From the current window, there are three result tabs for viewing. Click on the **Plotted Results** tab. Figure B.11 is a graph of altitude of the projectile versus time.

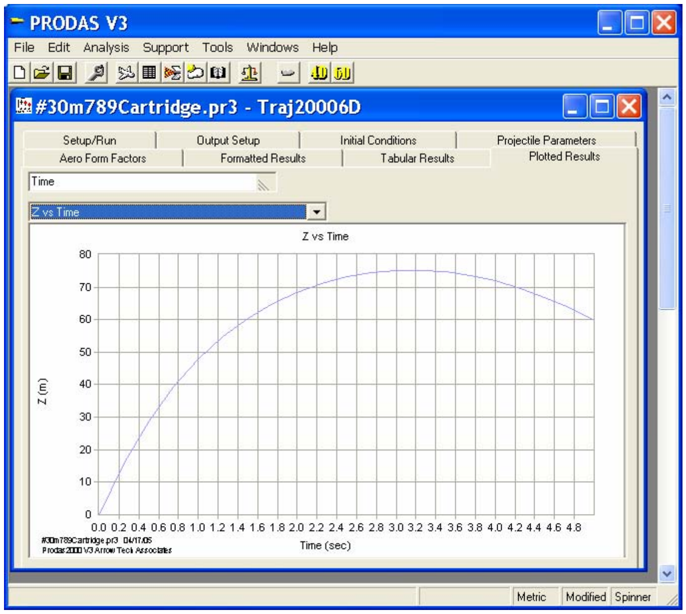


Figure B.11

To change the independent variable, click on the box next to **Time**. The dependant variable can also be changed by clicking on the box with the arrow next to **Z vs Time**. Next, click on the **Tabular Results** tab. A screen similar to Figure B.12 should appear.

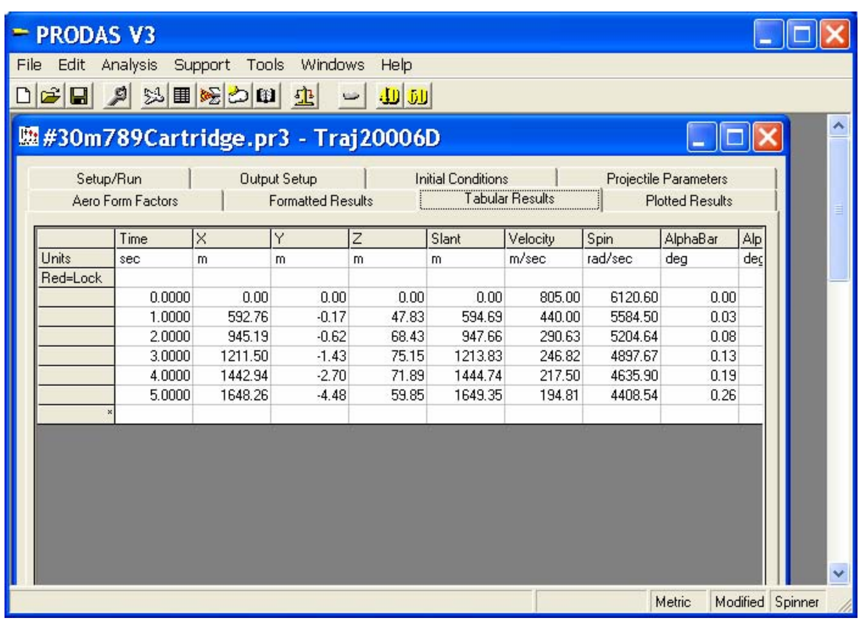


Figure B.12

This table reports the projectile characteristics at specified points in time. To change the time interval, click the **Output Setup** tab and rerun the simulation.

Now, click the **Formatted Results** tab. This is a summary report of conditions at time of fire and the conditions at the impact location of the projectile. Figure B.13 shows how the summary box should look on your screen.

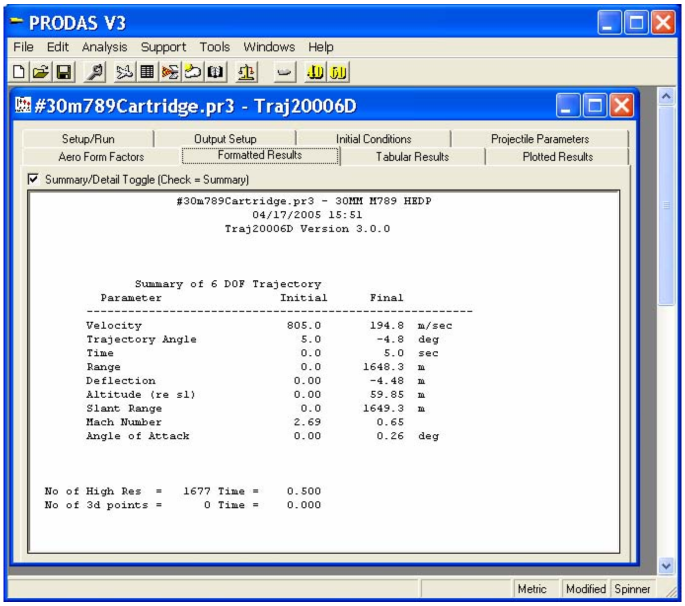


Figure B.13

The coordinate system in PRODAS is described in Figure B.14.

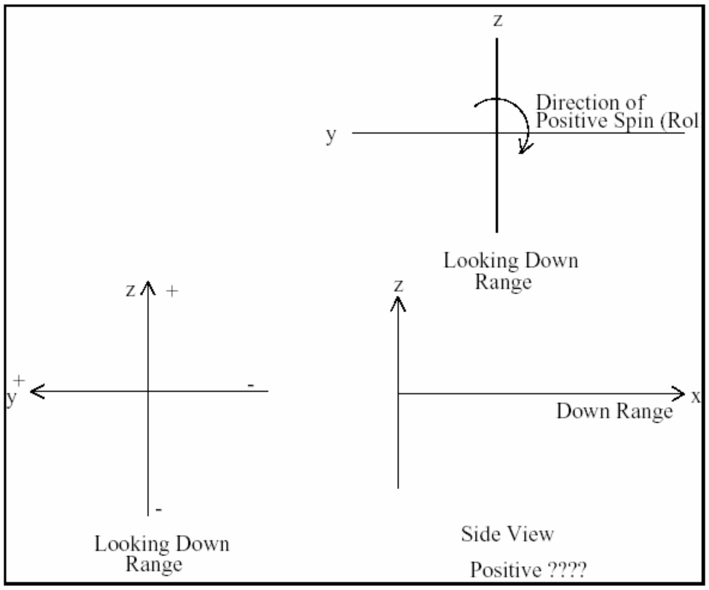


Figure B.14

B.6 SPECIAL CASES: MET TABLES AND MOVING AIRCRAFT

If the user wishes to input different meteorological conditions, a separate analysis must be run. This analysis should be run after the Stability Evaluation analysis. To open this analysis, click on the box illustrated with a white cloud under the **Support Menu**. When moving the computer's mouse over the boxes, a brief description will appear on your screen about each box as shown in Figure B.15.

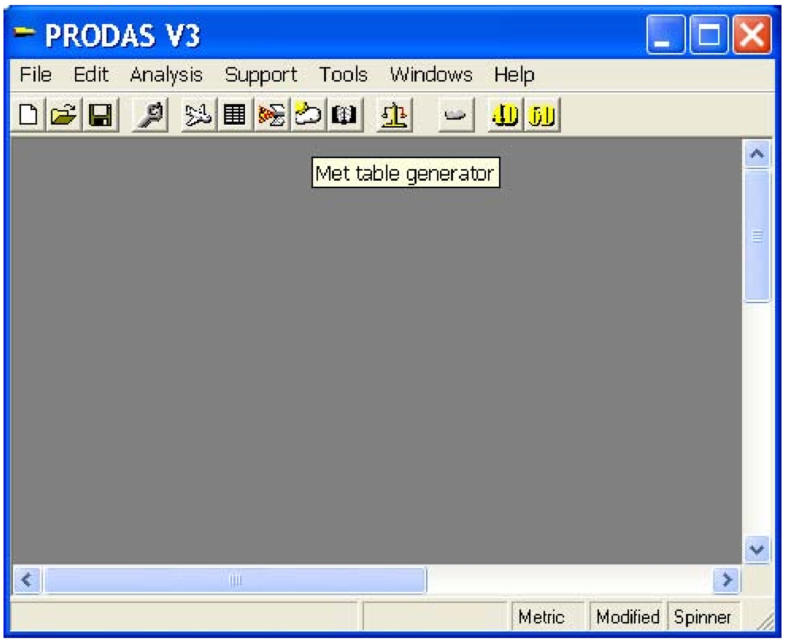


Figure B.15

Click on the box that says Met table generator. The box similar to Figure B.16 should appear on the screen:

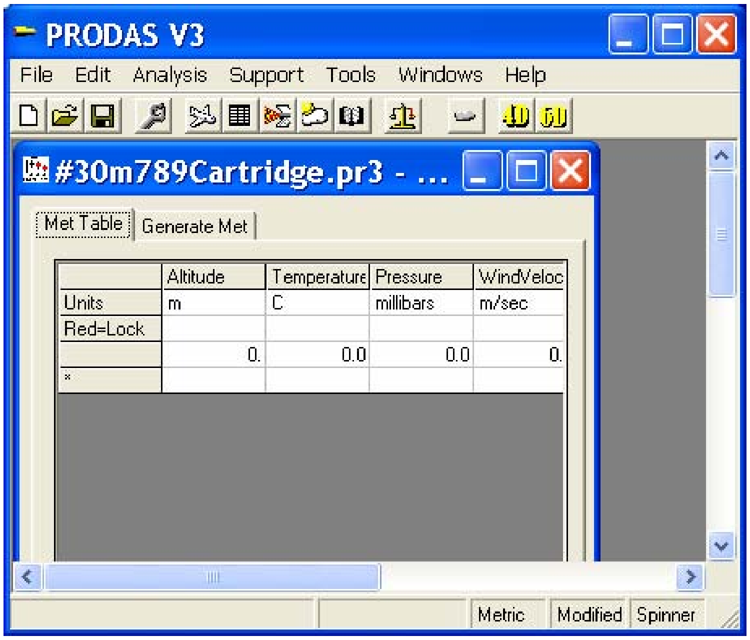


Figure B.16

Now click on the **Generate Met** tab. The screen should now look like Figure B.17.

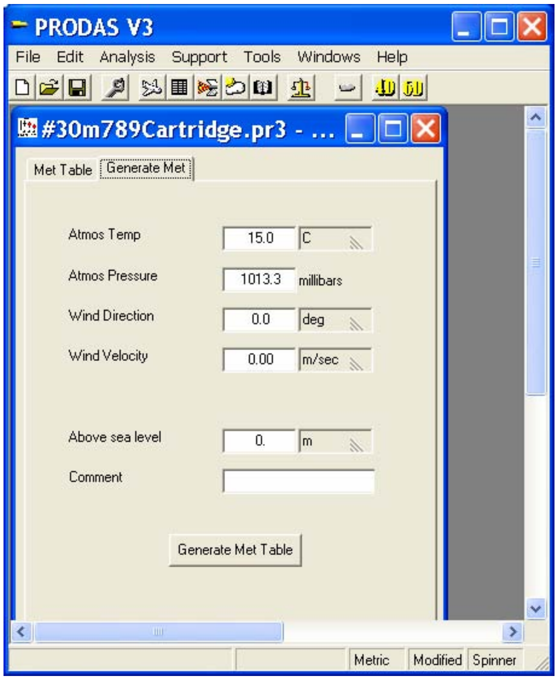


Figure B.17

Change the meteorological conditions to the desired values and click **Generate Met Table.** To use these new meteorological conditions the changeable parameter Met

Table Source must be changed to the User option. This parameter can be changed in the **Initial Conditions** tab of the trajectory simulation.

PRODAS simulations may also be run with moving aircraft. The aircraft's movement must be inputted into PRODAS after the Stability Evaluation Analysis. To set these parameters, click on the box illustrated with an airplane under the **Analysis Menu** as shown in Figure B.18.

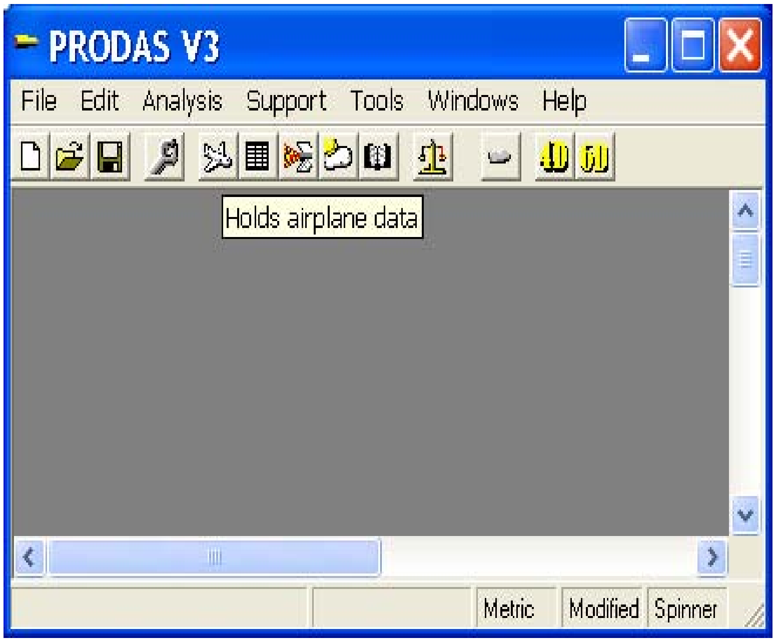


Figure B.18

After clicking this button, the screen should now look like Figure B.19.

Figure B.19

Make changes to any parameters necessary and minimize the box.

When doing a simulation with moving aircraft, both the 4DOF ad 6DOF trajectory simulations must be run. **Fixed Plane-4D** can be found under **Trajectories** in the **Analysis Menu**.

B.7 SCRIPT FILE AND PURPOSES

PRODAS V3 has the capability to allow the user to implement a visual basic (VB) script. The purpose of this script is to automate repetitive tasks, automate large scale investigations involving multiple projectiles and multiple analysis runs, direct attachment to WORD, and to allow unattended operation. In other words, a script file will automate the analyses and trajectory simulations previously mentioned in this appendix. The reader should be proficient in VB script. If this is not the case, there are many good textbooks on the subject such as [Ref. 11] and [Ref. 12]. The reader should also refer to the PRODAS V3 VB Script User Manual [Ref. 10]. The purpose of this explanation is to show the reader how to open a script file.

The first step is to start running the PRODAS software. Chose **Edit/Run VB Script** from the **Tools Menu**. Figure B.20 shows the box that should appear on the screen.

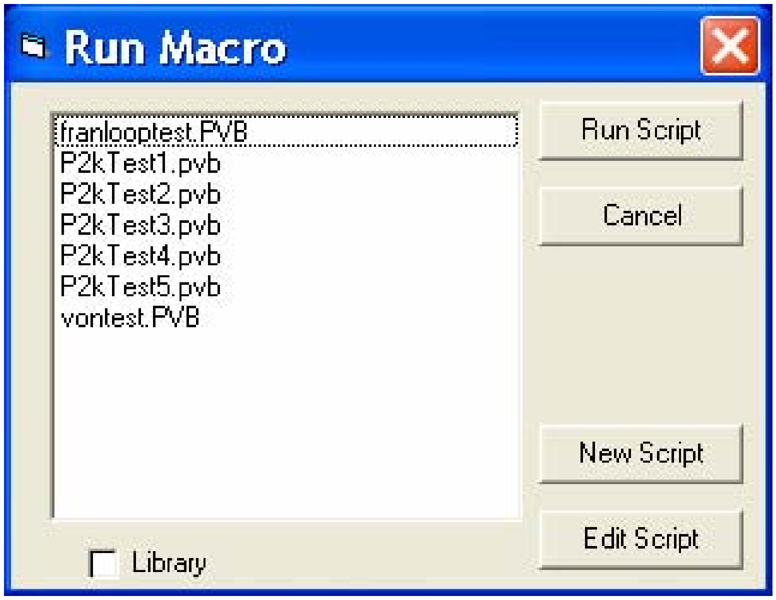


Figure B.20

The user has three options from this screen. To run a previously written script, highlight the script name and choose the **Run Script** button. To make changes to a previously written script, highlight the script name and choose the **Edit Script** button. To start a new script, choose **New Script** and a blank script page will appear.

APPENDIX C: USING ASCORE

The Accuracy Scoring (ASCORE) system is a LINUX based legacy program. It was written for the Yuma Test Center (YTC) for use in compiling the results of range firings. The program evaluates the accuracy of fired munitions by matching pulses to impacts. The program takes as an input the parameters of the specific firing (i.e. ammunition type, muzzle velocity, location of the target etc) and a series of documents containing the specifics of the test firing. These documents contain detailed information pertaining to the meteorological conditions, helicopter position, helicopter velocity, impact location and time of a round. ASCORE will then match each round fired with, according to a hardwired matching algorithm, the most likely match for a corresponding round. The results of this can be viewed in a compact report aptly named ASCORE report.

Since ASCORE is a command line based program within LINUX, some basic knowledge of the how to navigate and execute commands within the program is required. To start, the computer loaded with the ASCORE software should be turned on, and when prompted, a valid user name and password should be entered (the user name and pass must be obtained from YTC). This will allow the computer to finish loading and display a desktop similar to that of a windows environment. To bring up the command line prompt required to access the ASCORE program, a “shell” must first be opened on the desktop. On the taskbar at the bottom of the desktop, there will be four symbols on the left hand side. The user should click once on the symbol with a computer monitor with a sea shell in front of it (this should be the third from the left). This will bring up the ASCORE command line window (shown in Figure C.1).

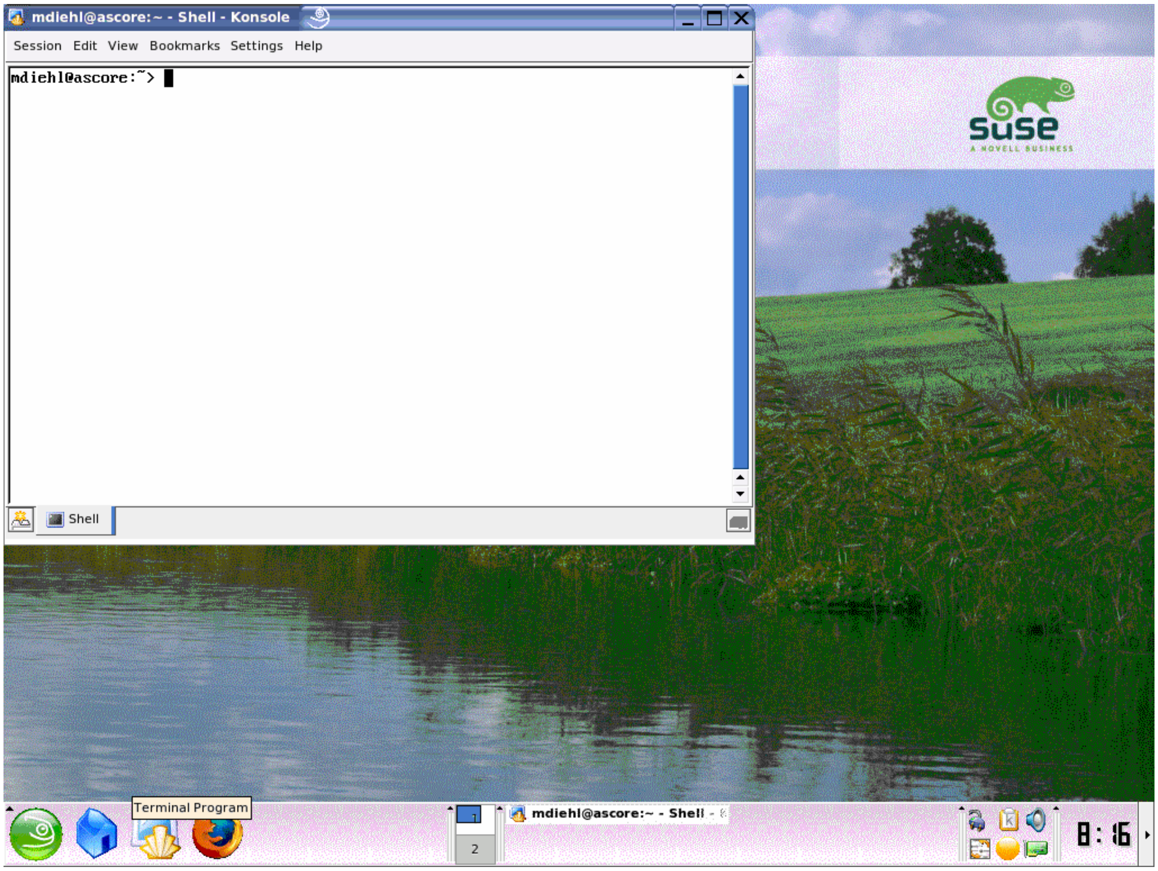


Figure C.1

Figure C.1 shows the basic command window used for operating ASCORE on a LINUX system.

The basic commands needed to navigate in the command line environment are listed below. The first three are standard DOS command line commands. The last two are specific to ASCORE. Figure A.2 illustrates the use of these commands.

- dir (directory, this will display the directory of your current location)
- cd (Change Directory)
- .. (Two periods will "back" you out of the current directory)
- kate (kate is the execution command and must precede any executable file)
- ascore (This is a special command used only to execute the ASCORE program)

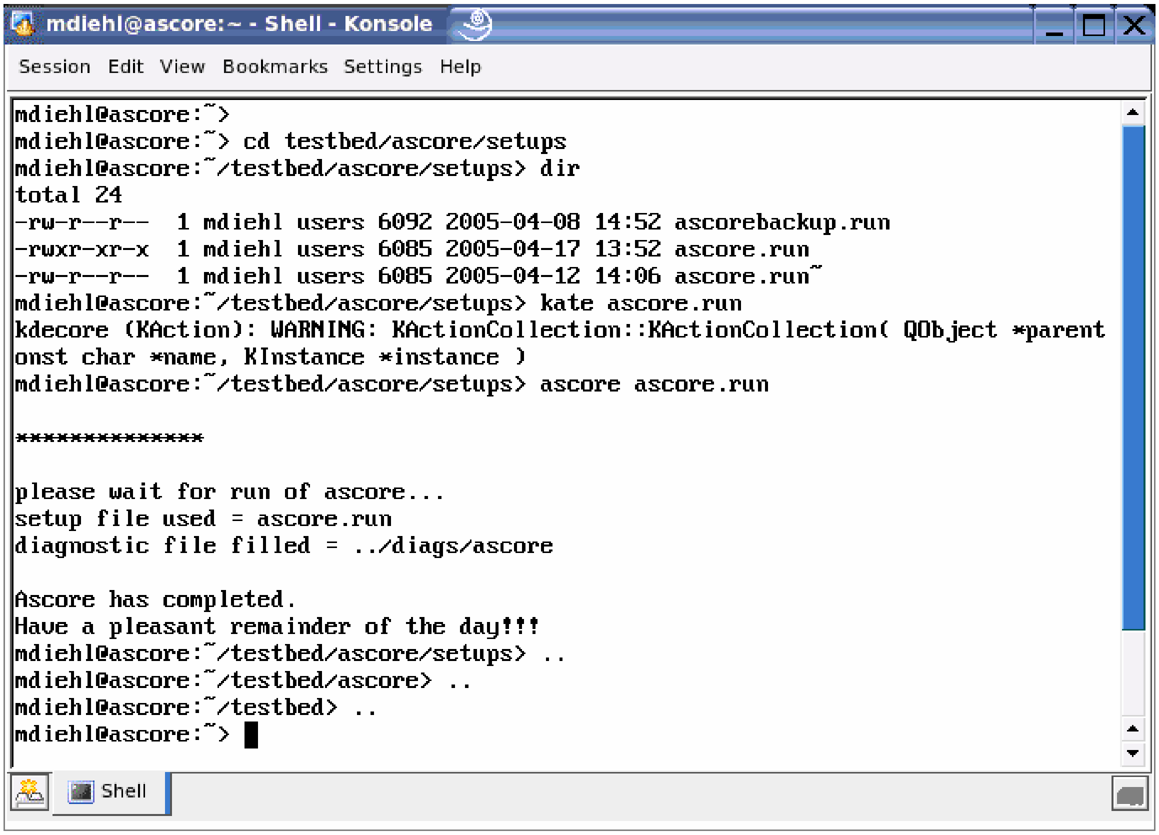


Figure C.2:

Figure C.2 shows an example of how each of the basic commands in ASCORE could be used. With these commands a user should be able to successfully navigate, open files , and execute programs in ASCORE.

The directory testbed/ascore contains the inputs and outputs for the ASCORE program; these are sorted into four main directories:

- diags
- inputs
- outputs
- setups

The directory testbed/ascore also contains a technical manual, ASCORE.HTM. An example of how to reach and execute this file is given in Figure C.3.

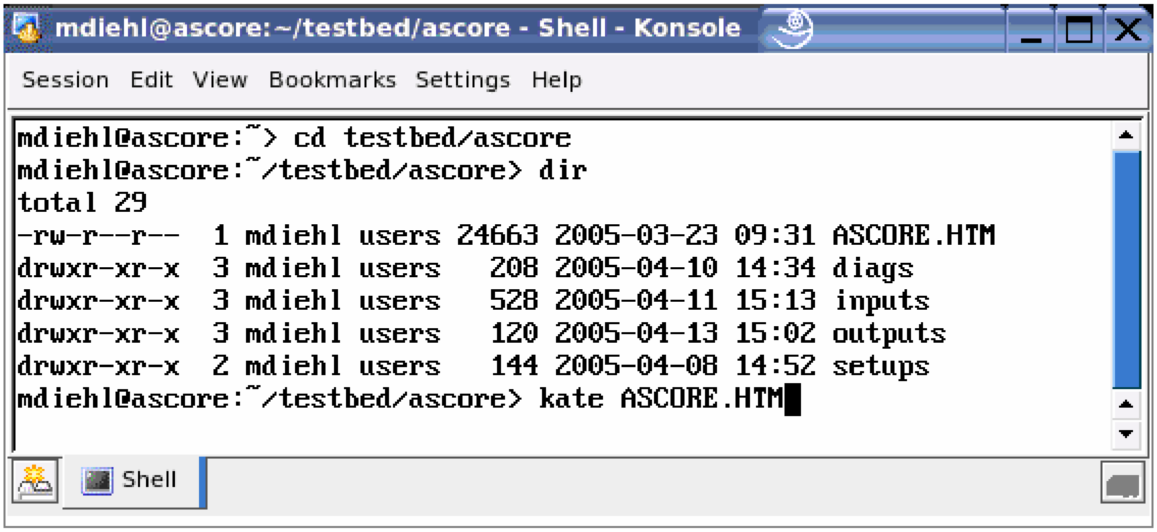
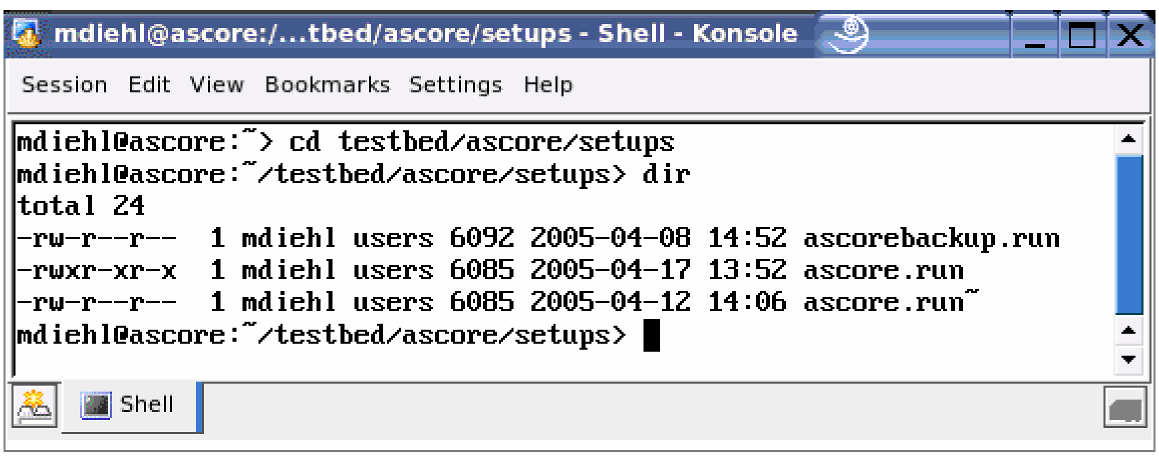


Figure C.3

Figure C.3 shows a listing of all of the directories that are pertinent to ASCORE.

C.1: SETUPS

The setups directory contains only one executable file named **ascore.run**. This file can be executed with both the kate and ascore commands. The file ascorebackup.run is a copy of the ascore.run file with its original configuration.



Fisgure C.4

Figure C.4 shows a list of all the files in the directory SETUPS.

C.1.1: ascore.run

The ascore.run file is where the majority of the parameters needed to run ASCORE are specified. In order to open this file for editing the kate command should be used. The following file is a sample ascore.run file. All print in bold are added comments and grey print shows all modules that should not be altered

$ASCORE START RUNSTREAM

Here the user must specify where the diagnostic file created during the execution of ASCORE will be written.

DIAGNOSTIC FILE (NO DEFAULT: MUST GIVE FILE)
[DIAG FILE = ../diags/ascore]

Each file in ASCORE requires a header so the program knows what it is opening. This is the header file and should not be changed.

STANDARD HEADER FILE (FOR REPORT HEADER) (OPTIONAL)
[STDHD FILE = ../inputs/stnd_hdr]

The user must specify the location of the burst time file here.

BURST TIME FILE (NO DEFAULT: MUST GIVE FILE)
[TIME FILE = ../inputs/first_pulse_estimate.timefile]

The user must specify the location of the TV or impact data here.

IMPACT INPUT SFF (NO DEFAULT: MUST GIVE FILE)
[IMPACT SFF = ../inputs/flt4.tv]

The following input modules allow the user to change the word that ASCORE uses to refer to Impact words (such as impact time and impact coordinates). Generally these will not need to be altered.

IMPACT INPUT TIME WORD (DEFAULT: TIME)
[IMPACT TIME WORD =]

IMPACT INPUT X WORD (DEFAULT: X)
[IMPACT X WORD =]

IMPACT INPUT Y WORD (DEFAULT: Y)
[IMPACT Y WORD =]

IMPACT INPUT Z WORD (DEFAULT: Z)
[IMPACT Z WORD =]

IMPACT INPUT DELTA X WORD (DEFAULT: DELTA X)

[IMPACT DELTA X WORD =]

IMPACT INPUT DELTA Y WORD (DEFAULT: DELTA Y)
[IMPACT DELTA Y WORD =]

This module allows the user to specify the location of the of the target in (x,y,z) coordinates. The coordinates listed below are for target 22 on the YPG range. All coordinates given are in reference to a point on the range called the IRCC. The location of the IRCC and the target can be on a map contained in Appendix A.

IMPACT VALUES (X,Y,Z) WRT THIS POINT WHICH IS WRT IRCC (METERS)
(SEPARATE WITH COMMAS) (DEFAULT: 0.,0.,0.)
[IMPACT RELATIVE SPOT = -1243.960295, 6983.710830, 29.828830]

This module is where the user can specify which MET towers were used and the location of the data file for each. The MET towers collect meteorological data during the test firing. If all three are not used, the program will not execute properly in its current configuration.

ENTER MET * (N) VALUES (N=8) FOR EACH MET TOWER SFF(s) PRESENT.

MET TOWER SFF(s) (OPTIONAL)
[MET SFF (8) = ../inputs/mettower08.sff]
[MET SFF (10) = ../inputs/mettower10.sff]
[MET SFF (12) = ../inputs/mettower12.sff]

The following input modules allow the user to change the word that ASCORE uses to refer to mettower data (i.e. air pressure, temperature, and wind speed to name a few). Generally these will not need to be altered.

MET TOWER TIME WORD (DEFAULT: TIME)
[MET TIME WORD (8) =]

MET TOWER AIR PRESSURE WORD (DEFAULT: PRESSURE)
[MET PRESSURE WORD (8) =]

MET TOWER A/C TEMPERATURE WORD (DEFAULT: TEMPERATURE L4)
[MET A/C TEMP WORD (8) =]

MET A/C WIND SPEED WORD (DEFAULT: HORZ WIND SP L4)
[MET A/C WIND SPEED WORD (8) =]

MET A/C WIND DIRECTION WORD (DEFAULT: HORZ WIND DR L4)
[MET A/C WIND DIR WORD (8) =]

MET TOWER TGT TEMPERATURE WORD (DEFAULT: TEMPERATURE L3)
[MET TGT TEMP WORD (8) =]

MET TGT WIND SPEED WORD (DEFAULT: HORZ WIND SP L3)
[MET TGT WIND SPEED WORD (8) =]

MET TGT WIND DIRECTION WORD (DEFAULT: HORZ WIND DR L3)
[MET TGT WIND DIR WORD (8) =]

The MET tower interpolation gap is the maximum allowable time between consecutive readings that ASCORE is allowed to interpolate between.

MET TOWER MAXIMUM INTERPOLATION GAP (DEFAULT: 5000 (MS))
[MET MAXGAP (8) =]

The default temperature for the MET data is used if a portion of the MET tower data is invalid. If this happens, the error is most likely the result of improper formatting or the absence of data.

TEMPERATURE USED IF MET DATA INVALID. 0-50 ALLOWED. (DEFAULT: 25 (DEGS C))
[NOMINAL TEMPERATURE =]

This is the default value for air pressure if readings are missing or the data is invalid.

AIR PRESSURE USED IF MET DATA INVALID. 980-1020 ALLOWED. (DEF: 1000 (MBARS))
[NOMINAL PRESSURE =]

***This module is only to be used if the target is moving. The user would specify the location of the file containing the location of the target.**

MOVING TARGET SFF (OPTIONAL) (IF STATIC, MOVING TARGET WORDS BELOW GO UNUSED)
[MOVER SFF =]

The following input modules allow the user to change the word that ASCORE uses to refer to a specified word. Generally these will not need to be altered

MOVING TARGET SFF TIME WORD (DEFAULT: TIME)

[MOVER TIME WORD =]

MOVING TARGET SFF VALIDITY WORD (DEFAULT: VALIDITY)
(PLEASE SET TO "UNUSED" IF NOT PRESENT IN SFF)
[MOVER VALIDITY WORD =]

MOVING TARGET SFF X WORD (DEFAULT: X POSITION)
[MOVER X WORD =]

MOVING TARGET SFF Y WORD (DEFAULT: Y POSITION)
[MOVER Y WORD =]

MOVING TARGET SFF Z WORD (DEFAULT: Z POSITION)
[MOVER Z WORD =]

***This value should not be altered**

MOVING TARGET MINIMUM VALIDITY (DEFAULT: 0)
(USE A NEGATIVE VALUE TO INDICATE MAXIMUM ALLOWED)
[MOVER VALIDITY =]

The target interpolation gap is the maximum allowable time between consecutive readings that ASCORE is allowed to interpolate between for a moving target.

MOVING TARGET MAXIMUM GAP FOR INTERPOLATION (DEFAULT: 5000
(MS))
[MOVER MAXGAP =]

This module allows the user to specify the type of ammunition used during a test firing.

AMMO TYPE. RKT (2.75 rocket), 20mma (20mm army), 20mmn (20mm navy), 30mm, 40mm, 105mm (DEFAULT: RKT)
[AMMO TYPE = 30mm]

The aircraft runin line is the angular line site from the aircraft to the target (i.e. if the target is due north from the aircraft the runin line would be 0.0 degrees and 180 degrees if the target was due south).

AIRCRAFT RUNIN LINE (DEGREES) (DEFAULT: 0.0)
[RUNIN LINE =]

This module allows the user to specify the initial speed of a weapon when fired from the aircraft.

MUZZLE VELOCITY (METERS/SEC) (NO DEFAULT. MUST BE PRESENT)
[MUZ VEL = 805.]

This module specifies where ASCORE will write the report file. The ASCORE report file is a compact version of the program results.

OUTPUT REPORT FILE (NO DEFAULT. MUST BE PRESENT)
[REPORT = ../outputs/report]

This module allows the user to specify have a pc file generated in the outputs folder. The pc file contains comma delimited data from the report with no header.

PC COMMA DELIMITERED FILE (OPTIONAL)
[PC FILE = ../outputs/pc]

This module specifies where ASCORE will write the impact diagnostic file. The impact file reports the specific results of the ASCORE round matching process, and will show any errors made while doing so.

IMPACT DIAGNOSTIC FILE (OPTIONAL)
[IMPACT DIAG = ../diags/impact]

This module specifies where ASCORE will write the pulse diagnostic file. The pulse file shows the pulses and times used for round matching and any errors made while doing so.

PULSE DIAGNOSTIC FILE (OPTIONAL)
[PULSE DIAG = ../diags/pulse]

This module is where the user specifies the location of the aircraft position data. The aircraft position data must be in (X,Y,Z) coordinates relative to the IRCC (Please see Appendix A for IRCC location specidifcs).

AIRCRAFT INPUT SFF (NO DEFAULT: MUST GIVE FILE)
[AIRCRAFT SFF = ../inputs/rte-223.sff]

The following input modules allow the user to change the word that ASCORE uses to refer to rte data (i.e. aircraft x,y,z position). Generally these will not need to be altered

AIRCRAFT INPUT TIME WORD (DEFAULT: TIME)
[AIRCRAFT TIME WORD =]

AIRCRAFT INPUT VALIDITY WORD (DEFAULT: VALIDITY)

(PLEASE SET TO "UNUSED" IF NOT PRESENT IN SFF)
[AIRCRAFT VALIDITY WORD = UNUSED]

AIRCRAFT INPUT X WORD (DEFAULT: X POSITION)
[AIRCRAFT X WORD =]

AIRCRAFT INPUT Y WORD (DEFAULT: Y POSITION)
[AIRCRAFT Y WORD =]

AIRCRAFT INPUT Z WORD (DEFAULT: Z POSITION)
[AIRCRAFT Z WORD =]

AIRCRAFT INPUT X VELOCITY WORD (DEFAULT: X VELOCITY)
[AIRCRAFT VX WORD =]

AIRCRAFT INPUT Y VELOCITY WORD (DEFAULT: Y VELOCITY)
[AIRCRAFT VY WORD =]

AIRCRAFT INPUT Z VELOCITY WORD (DEFAULT: Z VELOCITY)
[AIRCRAFT VZ WORD =]

AIRCRAFT MINIMUM VALIDITY (DEFAULT: 0)
(USE A NEGATIVE VALUE TO INDICATE MAXIMUM ALLOWED)
[AIRCRAFT VALIDITY =]

The aircraft interpolation gap is the maximum allowable time between consecutive readings that ASCORE is allowed to interpolate between.

AIRCRAFT MAXIMUM GAP FOR INTERPOLATION (DEFAULT: 5000 (MS))
[AIRCRAFT MAXGAP = 3000]

The processing mode tells ASCORE if telemetry (TM) data is being used. If TM data is used then the mode should be set to real. This requires an additional add on to the basic ASCORE program. The data currently collected at YPG utilizes the simulate mode.

PROCESSING MODE. SIMULATE or REAL (pulses) (NO DEFAULT: MUST GIVE VALUE)
[PMODE = SIMULATE]

The following modules are only utilized when TM data is used, and generally should not be changed.
TM INPUT SFF (NO DEFAULT: MUST GIVE VALUE IF REAL PULSES)
[TM SFF =]

TM TIME WORD (DEFAULT: TIME)
[TM TIME WORD =]

This module allows the user to specify the time delay from the beginning of the firing to the departure of the first round. If a value of zero is entered the program encounters a segmentation error and will not properly execute.

TIME (MS) FROM BURST START BEFORE BEGIN SIMULATING PULSES (DEFAULT: 0 (MS))
[START BIAS = 1000]

This module allows the user to input the firing rate of the gun being used. This rate will be used to compute the time between rounds when round matching (i.e. a rate 600 will fire a pulse every 100 ms).

SIMULATION SAMPLING RATE (PULSES/MIN) (DEFAULT: 600) (DEF YIELDS EVERY 100MS)
[SIM RATE = 600]

This module allows the user to specify how many pulses per burst to simulate.

NUMBER OF PULSES TO SIMULATE (1-50) (DEFAULT: 25)
[HOW MANY = 10]

This module will never need to be modified since they can only be used in conjunction with telemetry (TM) data.

DELAY REQUIRED BETWEEN REAL PULSES (MS) (DEFAULT: 75)
[DELAY BETWEEN =75]

REAL PULSE TYPE: FIRE OR BOLT (NO DEFAULT: MUST GIVE VALUE)
[PULSE TYPE =]

PULSE WORD(S) IN TM SFF. (NO DEFAULT: MUST GIVE VALUE)
(IF PULSE TYPE = BOLT, UP TO 4 WORDS CHECKED)
[TM PULSE WORD (1) =]

PULSE WORD BIT THAT GOES HIGH. (1-16,LEFT TO RIGHT)
(NO DEFAULT: MUST GIVE VALUE) (USED IF PULSE TYPE = FIRE)
[FIRE BIT =]

PULSE/ACCELEROMETER THRESHHOLD(S) TO EXCEED (DEFAULT: 500, OR PREVIOUS)
(ONE PER PULSE WORD IF PULSE TYPE = BOLT)

[THRESHHOLD (1) =]

This module allows the user to specify the maximum velocity that a weapon can achieve from the fire time to the impact time. For a pulse this will be the muzzle velocity.

MAXIMUM ROUND VELOCITY ALLOWED PER BURST (DEFAULT: 900., OR PREVIOUS)
(ONE PER BURST)
[MAXVEL (1) = 805]

The user can specify here if any of the burst data entered is not to be processed by ASCORE.

SKIP BURST PROCESSING OPTION (LEAVE BLANK TO PROCESS ROUND)
(ONE PER BURST. ANY VALUE WILL CAUSE A SKIP)
[SKIPB (1) =]
[SKIPB (2) =]
[SKIPB (3) =]
[SKIPB (4) =]
[SKIPB (5) =]
[SKIPB (6) =]
[SKIPB (7) =]
[SKIPB (8) =]
[SKIPB (9) =]
[SKIPB (10) =]
[SKIPB (11) =]
[SKIPB (12) =]
[SKIPB (13) =]
[SKIPB (14) =]
[SKIPB (15) =]
[SKIPB (16) =]

This module allows the user to specify specific pulses that are to be omitted from ASCORE.

SKIP PULSE PROCESSING OPTION (UP TO 50 PER BURST)
(SKIPP (N) = M IMPLIES SKIP PULSE M OF BURST N FROM TIME FILE)
[SKIPP (1) =]

This module allows the user to specify specific impacts that are to be omitted from ASCORE.

SKIP IMPACT PROCESSING OPTION (UP TO 50 PER BURST)
(SKIPI (N) = M IMPLIES SKIP IMPACT M OF BURST N FROM TIME FILE)

[SKIPI (1) =]

$ASCORE END RUNSTREAM

C.2 INPUTS

The inputs directory is where the majority of the data files required to run ASCORE are kept. The files in this directory are given in Figure C.5.

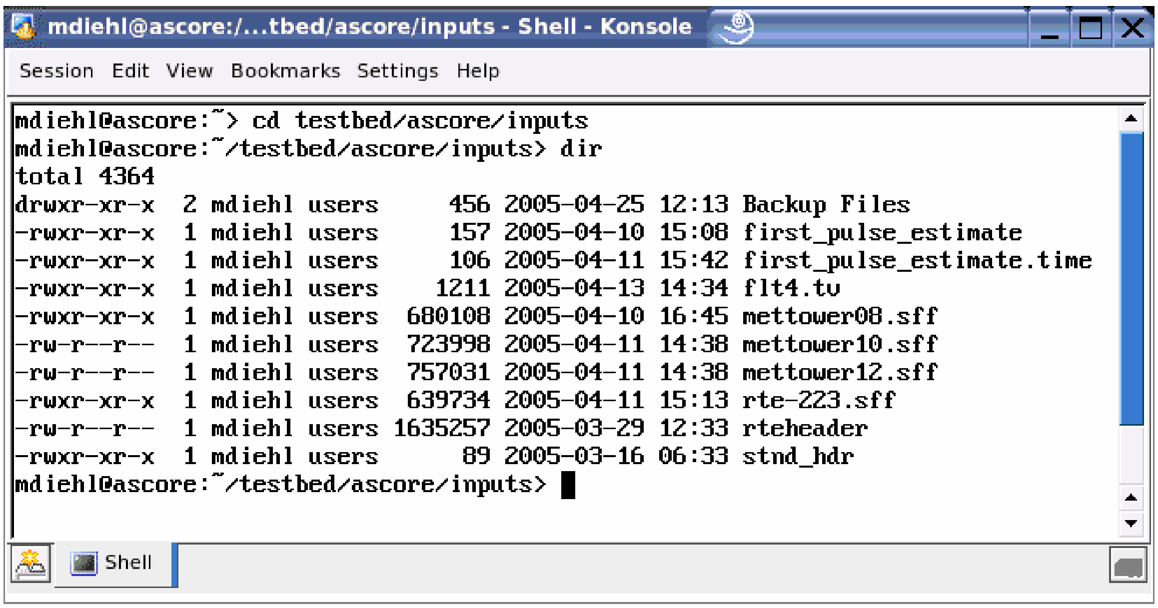


Figure C.5

Figure C.5 is listing of all the files in the INPUTS directory.

C.2.1 first_pulse_estimate

The first _pulse_estimate file tells ASCORE when the first pulse of the first burst is fired. This information is used to let ASCORE know the first firing sequence is initiated.

```
assume 30mm gun round rate 10Hz and 20 round bursts

pass #    hours      minutes       seconds milliseconds
-------------------------------------------

14              15  08    05                599
```

C.2.2 first_pulse_estimate.timefile

This file is where the user inputs the starting time and ending time for each burst in the test firing. The start time for each burst should be on the left and the end time on the right. The format for entering the start and end time is Julian (day # hour # minute # second # ms).

```
pass_14
    0#15#08#02#769#      0#15#08#07#  0#
pass_15
    0#15#09#15#929#      0#15#09#23#  0#
```

C.2.3 stnd_hdr

This is the standard header file, and should not be changed.

```
00 DEC 00   #00 FEB  01
30mm TESTXX        #00000      #CARLSON     #fXXXtXXX#YPGf00X
```

C.2.4 flt4.tv

This file is where the data for the impact of the fired ammunition is input. The coordinates for the impact location are relative to the center of the target (i.e the center of the target is (0,0,0))

The variables in this file are:

TIME: The time is the real time impact of the projectile in milliseconds.

JULIAN DAY: This is the day of the year (1-365) that the test occurred on, it only matters that the day is the same for all input data.

FLIGHT: The flight number is for sorting purposes only. It is only important that the flight number is consistent in all input files.

PASS: The pass number is the used to sort each round that is fired into its corresponding burst. ASCORE can process one or more burst at a time and it may be beneficial to take advantage of this by using multiple pass numbers in the same file. If only one burst of fire is being processed, then ensure that all pass numbers are the same for the input files.

X,Y,Z: The (x,y,z) are the Cartesian coordinates of the impact. The origin or reference point for this system is the center of the target.

DELTA X, DELTA Y: These parameters inform ASCORE if the projectile impacted the target. If an impact occurred then the value entered should be a 1 for both Delta X and Delta Y, zero otherwise.

As an example, the following values are used for flt4.tv.

TIME	JULIAN DAY	FLIGHT	PASS	X	Y	Z	DELTA X	DELTA Y
10.0lf	3d	4c	4c	11.3f	11.3f	11.3f	5d	5d
MILLISEC	NONE	NONE	NONE	METERS	METERS	METERS	COUNTS	COUNTS
54486746	343	224_	14__	-0.458	-0.17	-0.668	1	1
54560740	343	224_	15__	-0.422	-0.157	1.029	1	1

The flt4.tv file is comma delimited, starting with TIME and ending with TSU EL ERROR, corresponding with an input at the bottom of the file. Each variable has also been color coded with its corresponding input value.

```
11  11 "," "a"
08  DEC  2000        12  Apr  01          FLT  223  CENTER    XXXX-XXXX-XX
FRANKLIN        xxxxxx xxx
BEGIN PROG COM
END PROG COM
BEGIN USER COM
END USER COM
TV
TIME          ,JULIAN DAY     ,FLIGHT         ,PASS           ,X             ,Y
,Z          ,DELTA X        ,DELTA Y       ,TSU AZ ERROR    ,TSU EL ERROR
10.0lf          ,3d             ,4c            ,4c            ,11.3f         ,11.3f         ,11.3f
,5d           ,5d            ,12.8f          ,12.8f
MILLISEC        ,NONE              ,NONE              ,NONE              ,METERS
,METERS            ,METERS             ,COUNTS             ,COUNTS             ,RADIANS
,RADIANS
54558911.4, 343, 224_, 15__,  3,   -0.03,  0.37, 1, 1, 0, 0
54559060.9, 343, 224, 15, -3.03, 16.44, -1.49, 0, 0, 0, 0
```

C.2.5 rte-223.sff

The file rte-223.sff holds all the data on the position of helicopter conducting the test firing. It is important to ensure that the time entered at the beginning and end of this data file includes the times for all test firings and impacts. It is also important to ensure that the time intervals between each data entry are less than or equal to the interpolation gap that was entered in the ascore.run file. Failure to do this could result in a segmentation error.

The variables in this file are:

TIME: This time is used as a reference so ASCORE can match up helicopter location with the firing of the round.

Block ID, Validity, Retro, and RTE: These are bookkeeping devices and should generally be left at the default values.

X,Y,Z position: These are the locations of the helicopter in Cartesian coordinates with the IRCC as the origin or reference point (This can be seen visually on a map included in Appendix A).

X,Y,Z velocity: These are the velocities of the helicopter in the (x,y,z) directions.

X,Y,Z acceleration: These are the accelerations of the helicopter in the(x,y,z) directions.

As an example, the following values are used for rte-223.sff.

TIME	BLOCK ID	VALIDITY	RETRO ID	RTE	X POSITION	Y POSITION	Z POSITION
10.0lf	3d	1d	1d	3d	11.3f	11.3f	11.3f
MILLISEC	NONE	NONE	NONE	NONE	METERS	METERS	METERS
52870000	5	7	0	128	-1243.96	5983.71	229.82
52871000	5	7	0	128	-1243.96	5983.71	229.82

X VELOCITY	Y VELOCITY	Z VELOCITY	X ACCELERATION	Y ACCELERATION	Z ACCELERATION
10.3f	10.3f	10.3f	9.3f	9.3f	9.3f
M/SEC	M/SEC	M/SEC	M/SEC**2	M/SEC**2	M/SEC**2
14.793	-45.356	-5.362	0	0	0
10.375	-46.798	-5.33	0	0	0

The file rte-223.sff is comma delaminated, starting with TIME and ending with INST 9 COMP SOL, corresponding with an input at the bottom of the file. Each relevant variable has also been color coded with its corresponding input value. For ease of explanation, the file has been partially transferred into an excel file with each input below its associated unit and category. In practice, all input entries past Z acceleration will be zero.

32 32 "," "a"

BEGIN PROG COM
END PROG COM
BEGIN USER COM
END USER COM
RTE
TIME ,BLOCK ID ,VALIDITY STAGE ,RETRO ID ,RTE SOLUTION ,X POSITION ,Y POSITION ,Z POSITION ,X VELOCITY ,Y VELOCITY ,Z VELOCITY ,X ACCELERATION ,Y ACCELERATION ,Z ACCELERATION ,X RESIDUAL 200QD,Y RESIDUAL 200QD,Z RESIDUAL 200QD,X CONS EDT 200QD,Y CONS EDT 200QD,Z CONS EDT 200QD,X TOT EDT 200QD,Y TOT EDT 200QD,Z TOT EDT 200QD,INST 1 COMP SOL ,INST 2 COMP SOL ,INST 3 COMP SOL ,INST 4 COMP SOL ,INST 5 COMP SOL ,INST 6 COMP SOL ,INST 7 COMP SOL ,INST 8 COMP SOL ,INST 9 COMP SOL
10.0lf ,3d ,1d ,1d ,3d ,11.3f ,11.3f ,11.3f ,10.3f ,10.3f ,10.3f ,9.3f ,9.3f ,9.3f ,9.3f ,9.3f ,9.3f ,8d ,8d ,8d ,8d ,8d ,8d ,2d ,2d ,2d ,2d ,2d ,2d ,2d ,2d ,2d
MILLISEC ,NONE ,NONE ,NONE ,NONE ,METERS ,METERS ,METERS ,M/SEC ,M/SEC ,M/SEC ,M/SEC**2 ,M/SEC**2 ,M/SEC**2 ,METERS ,METERS ,METERS ,NONE ,NONE ,NONE ,NONE ,NONE ,NONE ,NONE ,NONE ,NONE ,NONE ,NONE ,NONE ,NONE ,NONE ,NONE
52870000, 5,7,0,128, -816.468, 9308.594, 225.863, 14.793, -45.356, -5.362, 0.000, 0.000, 0.000, 0.000, 0.000, 0.000, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0

C.2.6 Mettower08.sff

The met tower file contains all of the data pertaining to meteorological conditions on the range during the test firing. It is important to ensure that the time entered at the beginning and end of this data file encompasses the times for all test firings and impacts. It is also important to ensure that the time intervals between each data entry are less than or equal to the interpolation gap that were entered into the ascore.run file. Failure to do this could result in a segmentation error. There are three met tower files that need to be modified. Each of the three met tower files contains data for each of four levels on the tower. ASCORE takes the data from each level on all three towers and averages them. For the purposes of simulation, the same values for all four levels can be entered on each of the three towers. If data for a tower is not entered or is removed from the ascore.run file, then a runtime or segmentation error occurs.

The variables in this file are:

TIME: Time (in milliseconds) here marks the moment the corresponding meteorological conditions were recorded.

SYSTEM: The station number is the meteorological tower designator. This is an example for tower 8. The designator for towers 10 and 12 are 10 and 12.

TEMPERATURE: The temperature is the recorded temperature in degrees Celsius at the time of the observation.

WIND SPEED: This is the recorded speed of the wind in miles per hour at the time of the observation.

WIND DIRECCTION: This is angular direction the wind is coming from in radians.

PRESSURE: This is the recorded pressure in millibars at the time of the observation.

HUMIDITY: The humidity is the recorded water saturation level of the air in percent at the time of the observation.

As an example, the following values are used for mettower08.sff.

TIME	SYSTEM	STATION	TEMPERATURE	WIND SPEED	WIND DIR	PRESSURE	HUMIDITY
10.0lf	3d	3d	7.2f	9.4f	11.8f	10.5f	10.5f
MILLISEC	NONE	NONE	DEG C	M/SEC	RADIANS	MILLIBAR	PERCENT
52987800	4	8	8.2	2.1	0.593412	984	34

This file mettower08.sff is comma delaminated with each category, starting with TIME and ending with VERT WIND SP L4, corresponding with an input at the bottom of the file. Each relevant variable has also been color coded with its corresponding input value. For ease of explanation, the file has been partially transferred into an excel file with each input below its associated unit and category.

24 24 "," "a"
08 DEC 00 22 Jan 01 FLT 224 20MM TGT0YKJ9501 FRANKLIN xxxxxx
BEGIN PROG COM
SOURCE =
END PROG COM
BEGIN USER COM
END USER COM
MET
TIME ,SYSTEM ,STATION ,TEMPERATURE ,WIND SPEED ,WIND DIR ,PRESSURE ,HUMIDITY ,HORZ WIND SP L1 ,HORZ WIND DR L1 ,TEMPERATURE L1 ,VERT WIND SP L1 ,HORZ WIND SP L2 ,HORZ WIND DR L2 ,TEMPERATURE L2 ,VERT WIND SP L2 ,HORZ WIND SP L3 ,HORZ WIND DR L3 ,TEMPERATURE L3 ,VERT WIND SP L3 ,HORZ WIND SP L4 ,HORZ WIND DR L4 ,TEMPERATURE L4 ,VERT WIND SP L4
10.0lf ,3d ,3d ,7.2f ,9.4f ,11.8f ,10.5f ,10.5f ,9.4f ,11.8f ,7.2f ,9.4f ,9.4f ,11.8f ,7.2f ,9.4f ,9.4f ,11.8f ,7.2f ,9.4f ,9.4f ,11.8f ,7.2f ,9.4f
MILLISEC ,NONE ,NONE ,DEG C ,M/SEC ,RADIANS ,MILLIBAR ,PERCENT ,M/SEC ,RADIANS ,DEG C ,M/SEC ,M/SEC ,RADIANS ,DEG C ,M/SEC ,M/SEC ,RADIANS ,DEG C ,M/SEC ,M/SEC ,RADIANS ,DEG C ,M/SEC
52987800, 4, 8, 8.20, 2.1000, 0.59341192, 984.00000, 34.00000, 1.6000, 1.57079637, 14.40, 0.0000, 0.6000, 1.57079637, 16.40, 0.0000, 0.0000, 0.00000000, 16.30, 0.0000, 0.3606, 4.12438679, 17.20, 0.0000

C.3 OUTPUTS

The outputs directory holds the output files created by ASCORE after the execution of the program. An Output file (PC and Report) that is empty when opened is usually indicative of an error when ASCORE executed. The hold file shown in Figure C.6 is a directory used to backup files in the output directory.

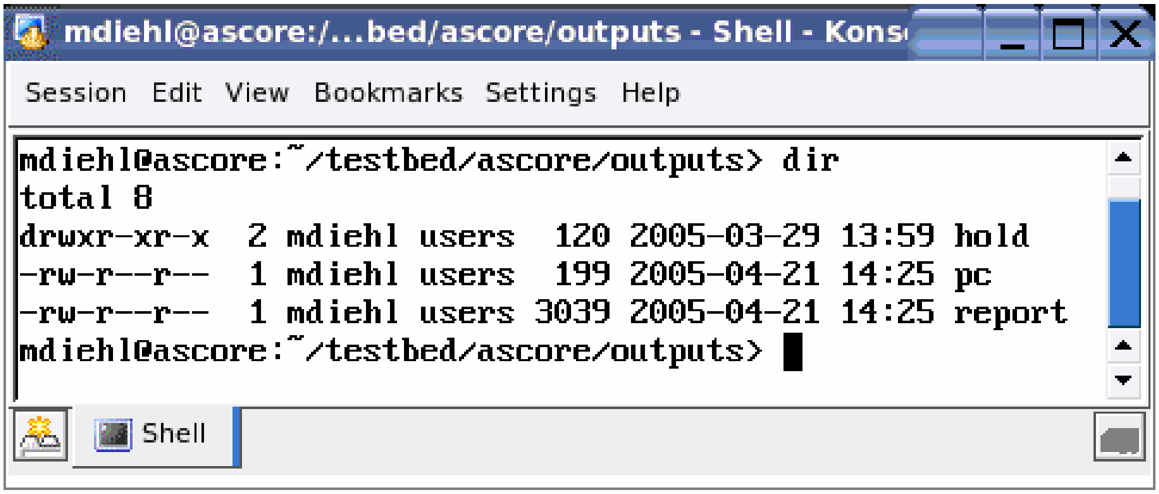
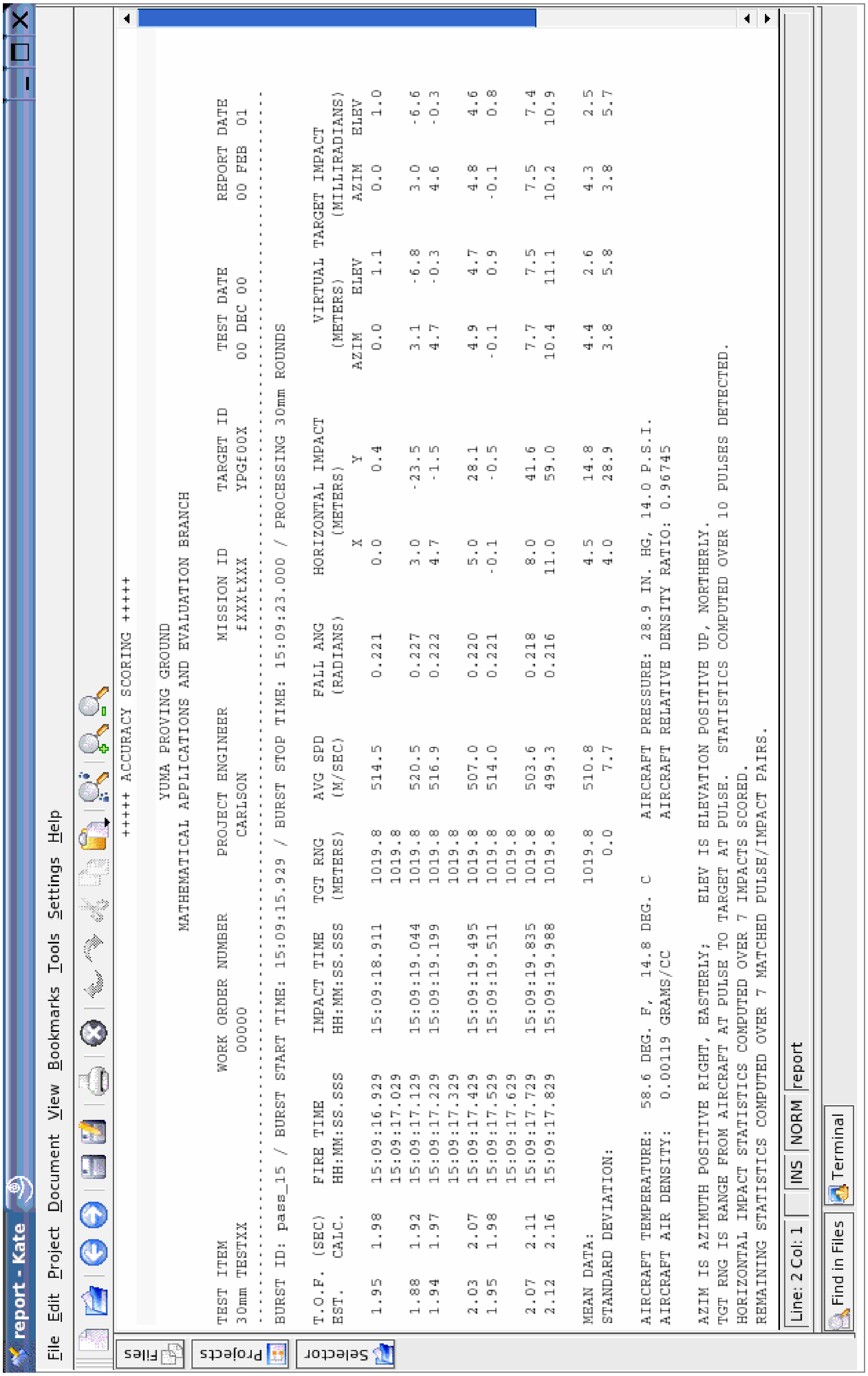


Figure C.6

Figure C.6 is a listing of all files in the outputs directory.

C.3.1 REPORT

The ASCORE report is the main output file. This document in Figure C.7 shows the results of the calculations ASCORE made for the virtual target impacts and round matching algorithm. The number of fire times will equal the input in the setups file for number of rounds to simulate. A time of flight and impact time next to a time of fire indicates that ASCORE has matched an impact that was entered into the flt4.tv file to a fire time. The number of pulses simulated and the number of impacts entered do not have to be the same. Summary statistics are then computed based on these calculations.



```
                              +++++ ACCURACY SCORING +++++

                                  YUMA PROVING GROUND
                        MATHEMATICAL APPLICATIONS AND EVALUATION BRANCH

TEST ITEM              WORK ORDER NUMBER    PROJECT ENGINEER     MISSION ID    TARGET ID    TEST DATE    REPORT DATE
30mm TESTXX                 00000               CARLSON            fXXXtXXX     YPGf00X     00 DEC 00    00 FEB  01
-------------------------------------------------------------------------------------------------------------------
BURST ID: pass_15 / BURST START TIME: 15:09:15.929 / BURST STOP TIME: 15:09:23.000 / PROCESSING 30mm ROUNDS

T.O.F. (SEC)  FIRE TIME     IMPACT TIME   TGT RNG  AVG SPD  FALL ANG  HORIZONTAL IMPACT      VIRTUAL TARGET IMPACT
EST.   CALC.  HH:MM:SS.SSS  HH:MM:SS.SSS  (METERS)  (M/SEC) (RADIANS)     (METERS)         (METERS)   (MILLIRADIANS)
                                                                         X        Y       AZIM   ELEV    AZIM   ELEV
 1.95   1.98  15:09:16.929  15:09:18.911   1019.8   514.5    0.221      0.0      0.4      0.0    1.1     0.0    1.0
              15:09:17.029                 1019.8
 1.88   1.92  15:09:17.129  15:09:19.044   1019.8   520.5    0.227      3.0    -23.5      3.1   -6.8     3.0   -6.6
 1.94   1.97  15:09:17.229  15:09:19.199   1019.8   516.9    0.222      4.7     -1.5      4.7   -0.3     4.6   -0.3
              15:09:17.329                 1019.8
 2.03   2.07  15:09:17.429  15:09:19.495   1019.8   507.0    0.220      5.0     28.1      4.9    4.7     4.8    4.6
 1.95   1.98  15:09:17.529  15:09:19.511   1019.8   514.0    0.221     -0.1     -0.5     -0.1    0.9    -0.1    0.8
              15:09:17.629                 1019.8
 2.07   2.11  15:09:17.729  15:09:19.835   1019.8   503.6    0.218      8.0     41.6      7.7    7.5     7.5    7.4
 2.12   2.16  15:09:17.829  15:09:19.988   1019.8   499.3    0.216     11.0     59.0     10.4   11.1    10.2   10.9

MEAN DATA:                                 1019.8   510.8               4.5     14.8      4.4    2.6     4.3    2.5
STANDARD DEVIATION:                           0.0     7.7               4.0     28.9      3.8    5.8     3.8    5.7

AIRCRAFT TEMPERATURE:  58.6 DEG. F,  14.8 DEG. C      AIRCRAFT PRESSURE: 28.9 IN. HG, 14.0 P.S.I.
AIRCRAFT AIR DENSITY:   0.00119 GRAMS/CC              AIRCRAFT RELATIVE DENSITY RATIO: 0.96745

AZIM IS AZIMUTH POSITIVE RIGHT, EASTERLY;    ELEV IS ELEVATION POSITIVE UP, NORTHERLY.
TGT RNG IS RANGE FROM AIRCRAFT AT PULSE TO TARGET AT PULSE.  STATISTICS COMPUTED OVER 10 PULSES DETECTED.
HORIZONTAL IMPACT STATISTICS COMPUTED OVER 7 IMPACTS SCORED.
REMAINING STATISTICS COMPUTED OVER 7 MATCHED PULSE/IMPACT PAIRS.
```

Figure C.7

Figure C.7 is a sample report file that ASCORE creates after it has completed processing the input data.Burst ID: This is the burst number that is being displayed.

Burst Start Time: This is the start time that was entered in the first_pulse_estimate.timefile.

Burst Stop Time: This is the stop time that was entered in the first_pulse_estimate.timefile.

T.O.F. EST: This is the estimated time of flight for a matched round

T.O.F. CALC: This is the calculated time of flight based on fire time and entered impact location.

TGT RANGE: This is the line of sight range from the helicopter to the target. For a non-moving helicopter this distance should not change from pulse to pulse. This distance can be calculated using $a \wedge 2 + b \wedge 2$ where a and b are the height above the target and the horizontal distance from the target respectfully.

AVERAGE SPEED: This is the average speed of the projectile from time of fire to time of impact.

FALL ANG: The fall angle is the angle the projectile makes with the ground at the time of impact. The fall angle is thewhat ASCORE uses to calculate a back trajectory and determine the virtual target impact.

HORIZONTAL IMPACT: This is the impact of the round in relative to the center of the target. If the runnin angle is 0.0 degrees then the values entered into the flt4.tv file for impacts should match the x and y impact location in the ASCORE report. If a runnin angle is used then the entered coordinates will be rotated by ASCORE in accordance with the runnin angle. Checking to ensure that the horizontal impacts match the values that were entered in the impact file is a very good idea and can help track down the source of potential errors.

VITUAL TARGET IMPACT: The virtual target impact for each projectile is based on the initial distance from the target and the fall angle. ASCORE has two approximations that it uses to calculate the back trajectory of impacted rounds to find the virtual target impact. If the initial distance is greater than 750 meters, then one type of calculation is used, if it is less than 750 than another type of calculation is used. There is no current documentation or YTC engineer that understands how these two calculations operate or why 750 meters was selected as a cutoff for the two algorithms.

C.3.2 pc

The pc report, shown if Figure C.8, is an unlabeled summary of some of the statistics in the ASCORE report.

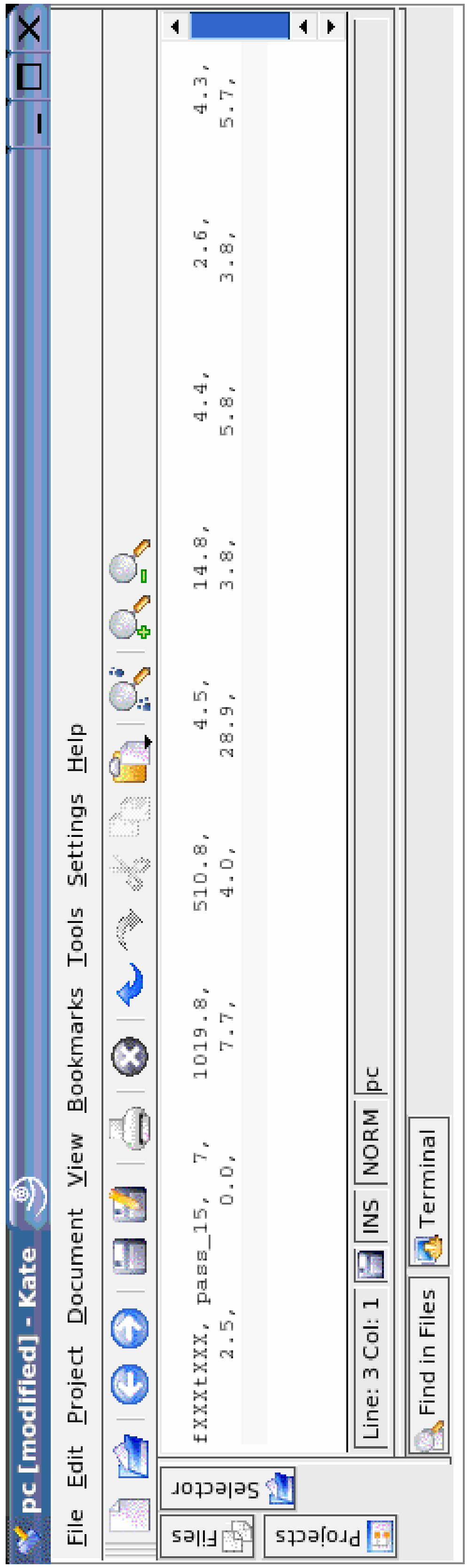

pc [modified] - Kate
File Edit Project Document View Bookmarks Tools Settings Help
fXXXtXXX, pass_15, 7, 1019.8, 510.8, 4.5, 14.8, 4.4, 2.6, 4.3,
2.5, 0.0, 7.7, 4.0, 28.9, 3.8, 5.8, 3.8, 5.7,
Line: 3 Col: 1
INS
NORM
pc
Find in Files
Terminal
Selector
Files
Projects

Figure C.8

Figure C.8 is a sample of the output of the pc file. All of the numbers reported in the pc file can be found at the bottom of the ASCORE report.

The following is a description of the variables shown in Figure C.8.

pass_15: This is the number that was assigned in the flt4.tv to identify the burst being looked at.

7: The 7 here is the number of pulses that were successfully matched with a fire time.

1019.8: This is the range of the helicopter from the target when the projectile was fired.

510.8: This number is the average speed in meters per second of all the matched pulses for the given burst.

4.5: This number is the average Horizontal X impact location in meters relative to the center of the target of all of the successfully matched pulses.

14.8: This number is the average Horizontal Y impact location in meters relative to the center of the target of all of the successfully matched pulses.

4.4: This number is the average azimuth of the given pulse in meters on the virtual target of all of the successfully matched pulses.

2.6: This number is the average elevation of the given pulse in meters on the virtual target of all of the successfully matched pulses

4.3: This number is the average azimuth of the given pulse in milliradianson the virtual target of all of the successfully matched pulses.

2.5: This number is the average elevation of the given pulse in milliradians on the virtual target of all of the successfully matched pulses

0.0: This is the standard deviation of the range of the helicopter for each matched pulse.

7.7: This number is the standard deviation of the speed, in meters per second, of all of the matched pulses in the given burst

4.0: This number is the standard deviation in Horizontal X impact location in meters relative to the center of the target of all of the successfully matched pulses.

28.9: This number is the standard deviation in Horizontal Y impact location in meters relative to the center of the target of all of the successfully matched pulses.

3.8: This number is the standard deviation in the azimuth of the given pulse in meters on the virtual target of all of the successfully matched pulses.

5.8: This number is the standard deviation of the elevation of the given pulse in meters on the virtual target of all of the successfully matched pulses

3.8: This number is the standard deviation of the azimuth of the given pulse in milliradians on the virtual target of all of the successfully matched pulses.

5.8: This number is the standard deviation of the elevation of the given pulse in milliradians on the virtual target of all of the successfully matched pulses

C.4 DIAGS

The diags directory, shown in Figure C.9, contains all of the diagnostic files created during the execution of ASCORE. These files contain some of the detail that is hidden in the report document.

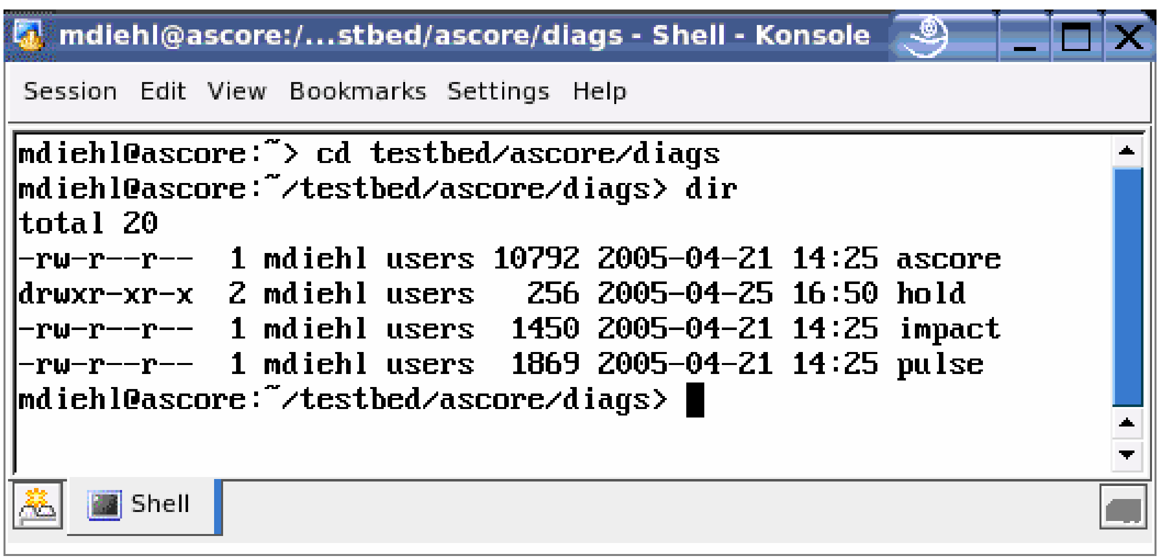


Figure C.9

Figure C.9 is a listing of all files in the diags directory.

C.4.1 ascore

The ascore file is the primary diagnostic file and is generally the first place that should be checked when an unexplained error occurs during the execution of ASCORE.

Below is an example of the ascore file. All print in bold type are added comments and not a part of the original file.

```
      ***** DIAGNOSTIC FILE for ascore    Wednesday, April 13, 2005
14:34  *****

      *** If desired, this file can be run back through ascore with a
single fix.
          Just change this file's name or the value in DIAG FILE.
```

This line shows the file that was executed in the setups directory. If multiple run files have been created ensure the proper file was executed.

```
      setup file used:  ascore.run
```

This line reports the output name and location for the asocre diagnostic file.

```
      [ DIAG FILE = ../diags/ascore ]  diagnostic output
```

This line reports where ASCORE retrieved the information for the standard header. This file should never be changed.

```
      [ STDHD FILE = ../inputs/stnd_hdr ] standard header file
```

This line reports the location of the file used for impact data.

```
      [ IMPACT SFF = ../inputs/flt4.tv ] impact input file
```

The following lines report if any changes were made in the setups file to words in ASCORE.

```
      IMPACT TIME WORD not in setup, so default will be used.
      [ IMPACT TIME WORD = TIME ] impact input time word
      IMPACT X WORD not in setup, so default will be used.
      [ IMPACT X WORD = X ] impact input x position word
      IMPACT Y WORD not in setup, so default will be used.
      [ IMPACT Y WORD = Y ] impact input y position word
      IMPACT Z WORD not in setup, so default will be used.
      [ IMPACT Z WORD = Z ] impact input z position word
      IMPACT DELTA X WORD not in setup, so default will be used.
      [ IMPACT DELTA X WORD = DELTA X ] impact delta x word
      IMPACT DELTA Y WORD not in setup, so default will be used.
      [ IMPACT DELTA Y WORD = DELTA Y ] impact delta y word
```

This line reports if ASCORE was successfully able to open and interpret the impact data.

```
      impact SFF opened okay.
```

This line reports the coordinates entered in the setups file for the target location.

```
      [ IMPACT RELATIVE SPOT = -1243.960,6983.711,29.829 ] impact SFF
wrt this x,y,z
```

The following lines report the location of the mettower files, any changes to the words used in the files, and if the files were opened ok.

```
      [ MET SFF (8) = ../inputs/mettower08.sff ] met tower 8 input SFF
```

MET TIME WORD (8) not in setup, so default will be used.
[MET TIME WORD (8) = TIME] met 8 time word
MET PRESSURE WORD (8) not in setup, so default will be used.
[MET PRESSURE WORD (8) = PRESSURE] met 8 air pressure word
MET A/C TEMP WORD (8) not in setup, so default will be used.
[MET A/C TEMP WORD (8) = TEMPERATURE L4] met 8 a/c temperature word
MET A/C WIND SPEED WORD (8) not in setup, so default will be used.
[MET A/C WIND SPEED WORD (8) = HORZ WIND SP L4] met 8 a/c wind speed word
MET A/C WIND DIR WORD (8) not in setup, so default will be used.
[MET A/C WIND DIR WORD (8) = HORZ WIND DR L4] met 8 a/c wind direction word
MET TGT TEMP WORD (8) not in setup, so default will be used.
[MET TGT TEMP WORD (8) = TEMPERATURE L3] met 8 tgt temperature word
MET TGT WIND SPEED WORD (8) not in setup, so default will be used.
[MET TGT WIND SPEED WORD (8) = HORZ WIND SP L3] met 8 tgt wind speed word
MET TGT WIND DIR WORD (8) not in setup, so default will be used.
[MET TGT WIND DIR WORD (8) = HORZ WIND DR L3] met 8 tgt wind direction word
MET MAXGAP (8) not in setup, so default will be used.
[MET MAXGAP (8) = 5000] met 8 max interpolation gap

met tower 8 SFF opened okay.

[MET SFF (10) = ../inputs/mettower10.sff] met tower 10 input SFF

MET TIME WORD (10) not in setup, so default will be used.
[MET TIME WORD (10) = TIME] met 10 time word
MET PRESSURE WORD (10) not in setup, so default will be used.
[MET PRESSURE WORD (10) = PRESSURE] met 10 air pressure word
MET A/C TEMP WORD (10) not in setup, so default will be used.
[MET A/C TEMP WORD (10) = TEMPERATURE L4] met 10 a/c temperature word
MET A/C WIND SPEED WORD (10) not in setup, so default will be used.
[MET A/C WIND SPEED WORD (10) = HORZ WIND SP L4] met 10 a/c wind speed word
MET A/C WIND DIR WORD (10) not in setup, so default will be used.
[MET A/C WIND DIR WORD (10) = HORZ WIND DR L4] met 10 a/c wind direction word
MET TGT TEMP WORD (10) not in setup, so default will be used.
[MET TGT TEMP WORD (10) = TEMPERATURE L3] met 10 tgt temperature word
MET TGT WIND SPEED WORD (10) not in setup, so default will be used.
[MET TGT WIND SPEED WORD (10) = HORZ WIND SP L3] met 10 tgt wind speed word
MET TGT WIND DIR WORD (10) not in setup, so default will be used.
[MET TGT WIND DIR WORD (10) = HORZ WIND DR L3] met 10 tgt wind direction word
MET MAXGAP (10) not in setup, so default will be used.

[MET MAXGAP (10) = 5000] met 10 max interpolation gap

met tower 10 SFF opened okay.

[MET SFF (12) = ../inputs/mettower12.sff] met tower 12 input SFF

MET TIME WORD (12) not in setup, so default will be used.
[MET TIME WORD (12) = TIME] met 12 time word
MET PRESSURE WORD (12) not in setup, so default will be used.
[MET PRESSURE WORD (12) = PRESSURE] met 12 air pressure word
MET A/C TEMP WORD (12) not in setup, so default will be used.
[MET A/C TEMP WORD (12) = TEMPERATURE L4] met 12 a/c temperature word
MET A/C WIND SPEED WORD (12) not in setup, so default will be used.
[MET A/C WIND SPEED WORD (12) = HORZ WIND SP L4] met 12 a/c wind speed word
MET A/C WIND DIR WORD (12) not in setup, so default will be used.
[MET A/C WIND DIR WORD (12) = HORZ WIND DR L4] met 12 a/c wind direction word
MET TGT TEMP WORD (12) not in setup, so default will be used.
[MET TGT TEMP WORD (12) = TEMPERATURE L3] met 12 tgt temperature word
MET TGT WIND SPEED WORD (12) not in setup, so default will be used.
[MET TGT WIND SPEED WORD (12) = HORZ WIND SP L3] met 12 tgt wind speed word
MET TGT WIND DIR WORD (12) not in setup, so default will be used.
[MET TGT WIND DIR WORD (12) = HORZ WIND DR L3] met 12 tgt wind direction word
MET MAXGAP (12) not in setup, so default will be used.
[MET MAXGAP (12) = 5000] met 12 max interpolation gap

met tower 12 SFF opened okay.

This is the default value used for temperature if it was not entered correctly in the setup and mettower data.
NOMINAL TEMPERATURE not in setup, so default will be used.
[NOMINAL TEMPERATURE = 25.000] temperature to use if no valid met data

This is the default value used for pressure if it was not entered correctly in the setup and mettower data.
NOMINAL PRESSURE not in setup, so default will be used.
[NOMINAL PRESSURE = 1000.000] air pressure to use if no valid met data

This line reports if the target was a moving target.
MOVER SFF not in setup; static target processing in effect.

This line reports the type of ammunition entered in the setup file.
[AMMO TYPE = 30mm] ammo type fired

This line reports the runin line angle entered in the setup file.
RUNIN LINE not in setup, so default will be used.

[RUNIN LINE = 0.00000] run in line

This line shows the result of the x and y coordinates rotation based on the runin line. If the runin line is 0.0 then the rotation does not change the x and y coordinates.

impact pos, a/c tspi, tgt tspi & met tower pos,wind components will be
rotated to runin line of 0.00000 degrees.
post rotated impact x,y = -1243.960 6983.711

These lines report where the respective output files from ASCORE are recorded and then reports if ASCORE successfully accessed them.

[REPORT = ../outputs/report] output report

[PC FILE = ../outputs/pc] optional comma delimitered pc file

[IMPACT DIAG = ../diags/impact] optional impact diagnostic file

[PULSE DIAG = ../diags/pulse] optional pulse diagnostic file

attempting to open requested output file(s)...
opens of output files went okay.

This line reports where ASCORE took the rte-223.sff (aircraft) data from. This was entered in the setups file

[AIRCRAFT SFF = ../inputs/rte-223.sff] a/c input file

These lines report any changes made in words used to reference the aircraft or the aircraft position.

AIRCRAFT TIME WORD not in setup, so default will be used.
[AIRCRAFT TIME WORD = TIME] aircraft input time word
[AIRCRAFT VALIDITY WORD = UNUSED] aircraft input validity word
AIRCRAFT X WORD not in setup, so default will be used.
[AIRCRAFT X WORD = X POSITION] aircraft input x position word
AIRCRAFT Y WORD not in setup, so default will be used.
[AIRCRAFT Y WORD = Y POSITION] aircraft input y position word
AIRCRAFT Z WORD not in setup, so default will be used.
[AIRCRAFT Z WORD = Z POSITION] aircraft input z position word
AIRCRAFT VX WORD not in setup, so default will be used.
[AIRCRAFT VX WORD = X VELOCITY] aircraft input x velocity word
AIRCRAFT VY WORD not in setup, so default will be used.
[AIRCRAFT VY WORD = Y VELOCITY] aircraft input y velocity word
AIRCRAFT VZ WORD not in setup, so default will be used.
[AIRCRAFT VZ WORD = Z VELOCITY] aircraft input z velocity word
aircraft input SFF opened ok.

since no aircraft validity word, will not check AIRCRAFT VALIDITY.

This line reports the interpolation gap entered in the setups file for aircraft position.

[AIRCRAFT MAXGAP = 3000] aircraft maximum interpolation gap

This line reports if telemetry or simulate mode was used.

[PMODE = SIMULATE] processing mode: SIMULATE pulses or REAL pulses from tm.

This line reports the start bias entered in the setups file.

[START BIAS = 1000] delay from burst start before simulated pulses begin

This line reports the rate that was entered for pulse generation in the setups file.

[SIM RATE = 600] simulated pulses per minute

This line reports how many pulses ASCORE was asked to generate in the setups file.

[HOW MANY = 10] number of pulses simulated

The pass id lines give the start and end time for each pass entered in the first_pulse_estimate.timefile.

pass id: pass_14

Start: 0 15:08:02.769 54482769 ms Stop : 0 15:08:07.000 54487000 ms

pass id: pass_15

Start: 0 15:09:15.929 54555929 ms Stop : 0 15:09:23.000 54563000 ms

number of passes = 2

This line reports the location of the file containing the burst start and stop times.

[TIME FILE = ../inputs/first_pulse_estimate.timefile] burst time file

This line reports the maximum velocity that was entered in the setups file.

[MAXVEL (1) = 805.0] round maximum velocity, burst: pass_14

MAXVEL (2) not in setup, so previous will be used.

[MAXVEL (2) = 805.0] round maximum velocity, burst: pass_15

This line reports any burst that was skipped and subsequently not processed due to a request made in the setups file.

[SKIPB (1) =] processing option, non-blank skips burst: pass_14

[SKIPB (2) =] processing option, non-blank skips burst: pass_15

These lines reports any particular pulses that were not processed from a specified burst due to a request made in the setups file.

SKIPP (N) = M. do not process pulse M of burst N listed below...(if any)

SKIPI (N) = M. do not process impact M of burst N listed below...(if any)

These lines report the success of each processed burst.

********PROCESSING BURST: pass_14...

no impacts found...will skip burst

********processing not successful for burst: pass_14. see above for details.

********PROCESSING BURST: pass_15...
RAN OUT OF IMPACT DATA

Now recovering tower coordinates for MET 08 and rotating to 0.000 degrees...

Survey requested: name = MET.TOWER.9 date = 0 time = 0
Survey used: file = /usr/local/coords/MET.TOWER.9_1998144_1100
Survey had no targets present.

Now recovering tower coordinates for MET 10 and rotating to 0.000 degrees...

Survey requested: name = MET.TOWER.10 date = 0 time = 0
Survey used: file = /usr/local/coords/MET.TOWER.10_1998144_1100
Survey had no targets present.

Now recovering tower coordinates for MET 12 and rotating to 0.000 degrees...

Survey requested: name = MET.TOWER.12.STR date = 0 time = 0
Survey used: file = /usr/local/coords/MET.TOWER.12.STR_1998144_1100
Survey had no targets present.
using MET 10 data
a/c & tgt temperature used: 14.8 13.6 (degrees c)
air pressure used: 996.0 (millibars)
a/c & tgt wind direction used: 2.778 2.679 (radians)
a/c & tgt wind speed used: 2.247 2.795 (m/sec)
PROCESSING 30mm ROUNDS

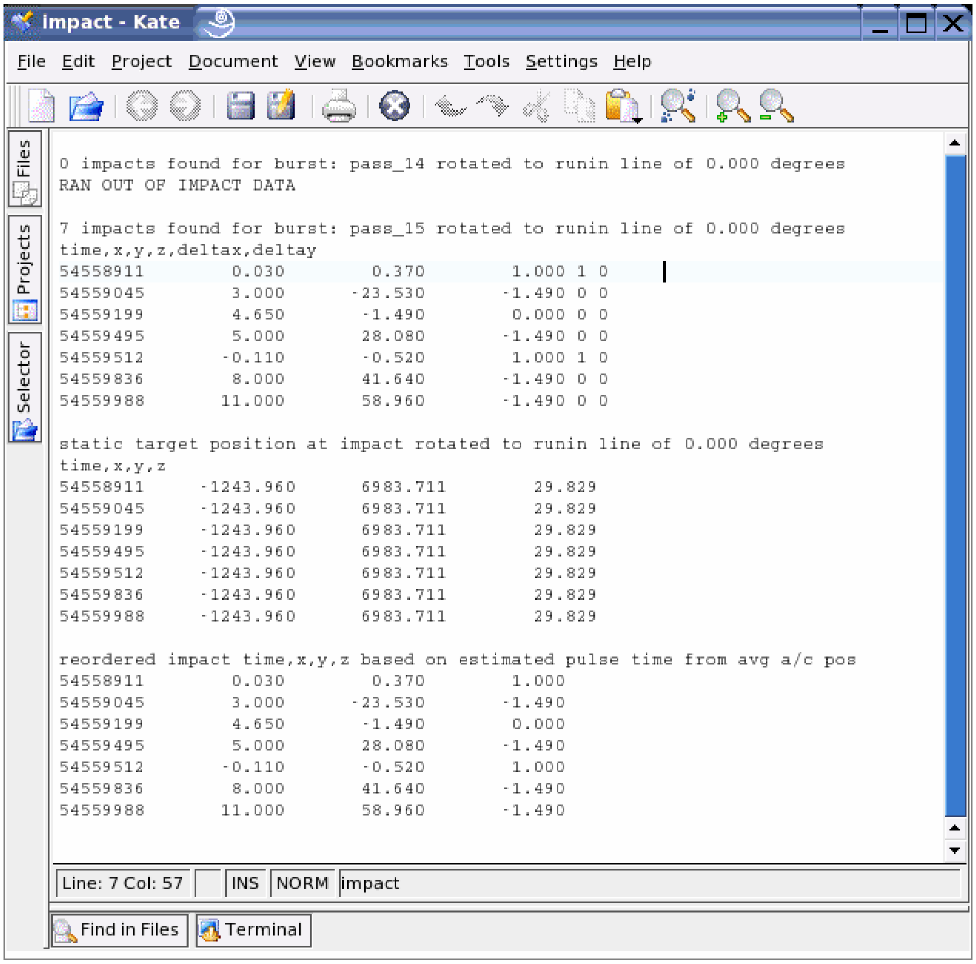
This report shows how ASCORE matched up the impacts entered with the pulses generated.
PULSES NUMBERED FROM 1 TO 10. IMPACTS NUMBERED FROM 1 TO 7...
pulse 1 is matched with impact 1
pulse 2 is unmatched.
pulse 3 is matched with impact 2
pulse 4 is matched with impact 3
pulse 5 is unmatched.
pulse 6 is matched with impact 4
pulse 7 is matched with impact 5
pulse 8 is unmatched.
pulse 9 is matched with impact 6
pulse 10 is matched with impact 7

matched pulse/impact info :
aircraft and impact x,y,z in virtual target coordinate system
fall angle wrt horiz at impact, x,z,az,el miss
-0.000 -1019.803 0.000 0.030 0.167 1.053
0.221 0.030 1.057 0.029 1.037
-0.000 -1019.803 0.000 3.000 -22.781 -6.075

```
          0.227       3.069       -6.768       3.009      -6.636
         -0.000   -1019.803        0.000        4.650      -1.461       -
0.292
          0.222       4.657       -0.330       4.566      -0.323
         -0.000    -1019.803          0.000          5.000         27.827
4.046
          0.220       4.867        4.651       4.773       4.561
         -0.000    -1019.803          0.000         -0.110         -0.706
0.879
          0.221      -0.110        0.862      -0.108       0.845
         -0.000    -1019.803          0.000          8.000         41.124
6.705
          0.218       7.690        7.515       7.540       7.369
         -0.000    -1019.803          0.000         11.000         58.107
10.101
          0.216      10.407       11.103      10.205      10.887

      ********processing completed successfully for burst: pass_15

      ascore:  COMPLETED WITH NO ERRORS DETECTED.
```

C.4.2 impact

The impact file, shown in Figure C.10, reports the runin line coordinate translation for each burst. If the runin line is 0.0 degrees then the translation should not affect the coordinates. Both the target and helicopter locations are rotated to the runnin line. The target should be the same for all points unless it is a moving target. The last item reported in the impact file is a list of the impacts entered in the flt4.tv file arranged in ascending order based on impact time.



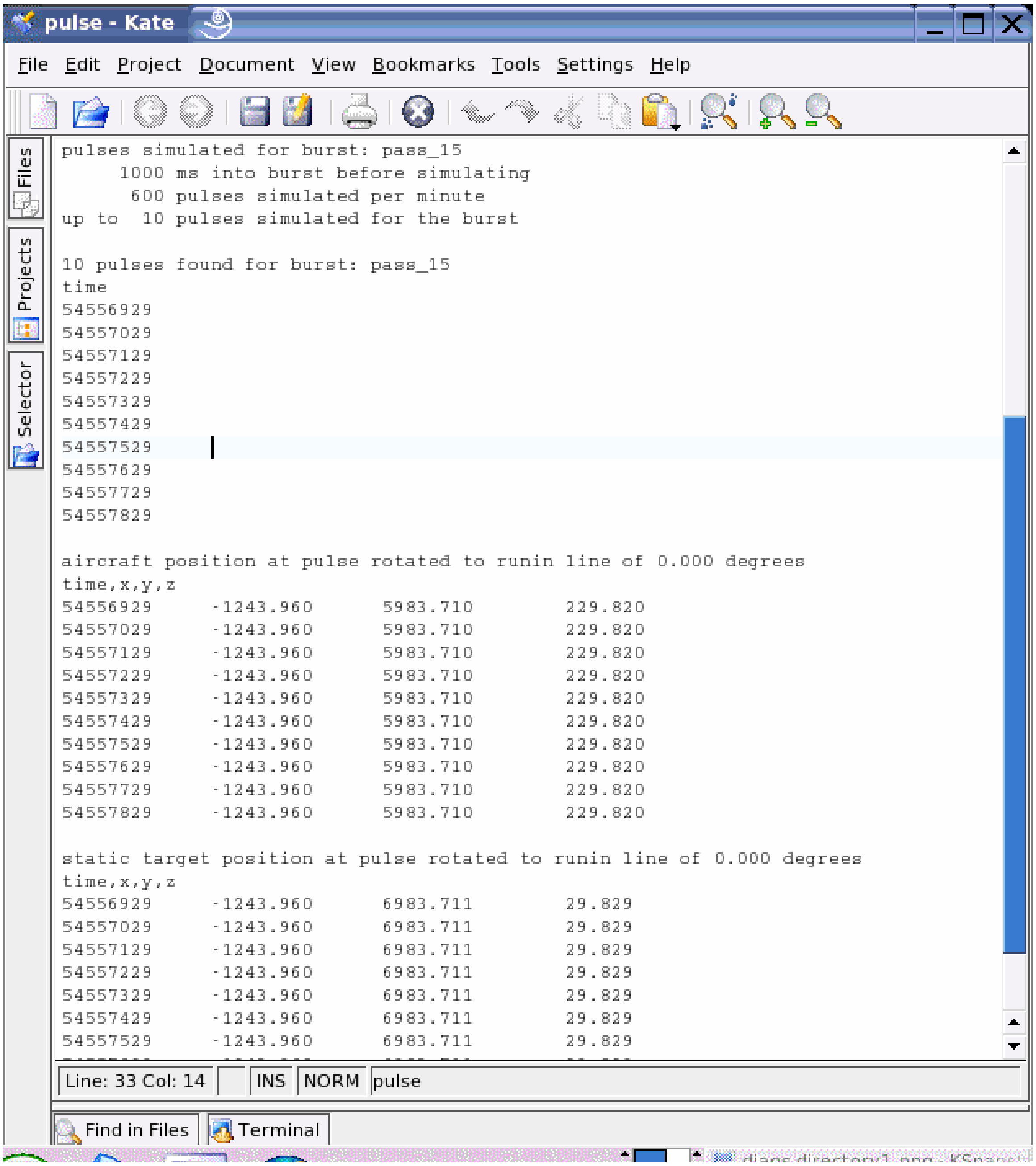

```
0 impacts found for burst: pass_14 rotated to runin line of 0.000 degrees
RAN OUT OF IMPACT DATA

7 impacts found for burst: pass_15 rotated to runin line of 0.000 degrees
time,x,y,z,deltax,deltay
54558911        0.030       0.370       1.000 1 0
54559045        3.000     -23.530      -1.490 0 0
54559199        4.650      -1.490       0.000 0 0
54559495        5.000      28.080      -1.490 0 0
54559512       -0.110      -0.520       1.000 1 0
54559836        8.000      41.640      -1.490 0 0
54559988       11.000      58.960      -1.490 0 0

static target position at impact rotated to runin line of 0.000 degrees
time,x,y,z
54558911     -1243.960     6983.711        29.829
54559045     -1243.960     6983.711        29.829
54559199     -1243.960     6983.711        29.829
54559495     -1243.960     6983.711        29.829
54559512     -1243.960     6983.711        29.829
54559836     -1243.960     6983.711        29.829
54559988     -1243.960     6983.711        29.829

reordered impact time,x,y,z based on estimated pulse time from avg a/c pos
54558911        0.030       0.370       1.000
54559045        3.000     -23.530      -1.490
54559199        4.650      -1.490       0.000
54559495        5.000      28.080      -1.490
54559512       -0.110      -0.520       1.000
54559836        8.000      41.640      -1.490
54559988       11.000      58.960      -1.490
```

Figure C.10

Figure C.10 shows a sample output of the impact file. Rotations made to the helicopter position and target location are made based on the runin line here. The impact times are also reordered in ascending order here.

C.4.3 pulse

The pulse file, shown in Figure C.11, is a summary of when each pulse was generated and the position of the helicopter after it has been rotated to the runin line at the time of fire for each pulse.




```
pulses simulated for burst: pass_15
     1000 ms into burst before simulating
      600 pulses simulated per minute
up to  10 pulses simulated for the burst

10 pulses found for burst: pass_15
time
54556929
54557029
54557129
54557229
54557329
54557429
54557529
54557629
54557729
54557829

aircraft position at pulse rotated to runin line of 0.000 degrees
time,x,y,z
54556929     -1243.960      5983.710       229.820
54557029     -1243.960      5983.710       229.820
54557129     -1243.960      5983.710       229.820
54557229     -1243.960      5983.710       229.820
54557329     -1243.960      5983.710       229.820
54557429     -1243.960      5983.710       229.820
54557529     -1243.960      5983.710       229.820
54557629     -1243.960      5983.710       229.820
54557729     -1243.960      5983.710       229.820
54557829     -1243.960      5983.710       229.820

static target position at pulse rotated to runin line of 0.000 degrees
time,x,y,z
54556929     -1243.960      6983.711       29.829
54557029     -1243.960      6983.711       29.829
54557129     -1243.960      6983.711       29.829
54557229     -1243.960      6983.711       29.829
54557329     -1243.960      6983.711       29.829
54557429     -1243.960      6983.711       29.829
54557529     -1243.960      6983.711       29.829
```

Figure C.11

Figure C.11 is a sample output for the pulse file. The pulse file reports the coordinates of the target and the helicopter after they have been rotated to the runnin line. The file also reports the number of rounds it found in the flt4.tv file. A good error check for the flt4.tv file is to ensure the number of pulses found in the pulse file equals the number entered in the flt4.tv file.

C.5 Miscellaneous Information

C.5.1 Transferring files from ASCORE to windows

During the course of manipulating the ASCORE program it is necessary to generate very large files that would be impractical to type by hand. It can be useful to transfer a template of the file you wish to manipulate from Linux over to windows for use in Excel or Word and then transfer it back. One way to do this is to use a thumb drive that plugs into the USB port of the Linux computer. Once plugged into the USB port of the Linux system the drive should appear under my computer (see Figure C.12).

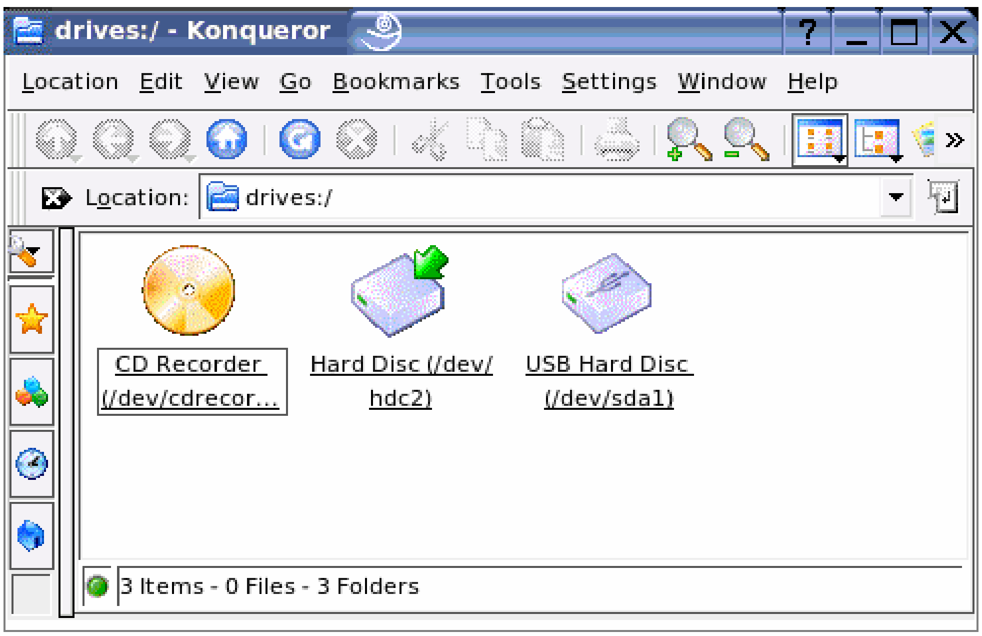


Figure C.11

Figure C.11 shows the drives available on a Linux system with a USB external memory device correctly attached.

Occasionally Linux will tell the user it is not authorized to write to a driver device when attempting to write to and from the USB drive. If this occurs all that needs to be done is to copy a file form the drive to Linux first and then there should not be any problem writing to the thumb drive.

Most of the files that need to be manipulated in the ASCORE program have 3 parts to them. These parts are the file header, data labels, and then the actual data. These files are often confusing to read because they are comma delimited in the ASCORE program. It is useful to take this format and transfer it to excel in a column delimited format that is conducive to the type of file manipulation that needs to be done. Once the file needing to be changed is on the windows system, simply open excel and select import data under the data tab (see Figure C.12).

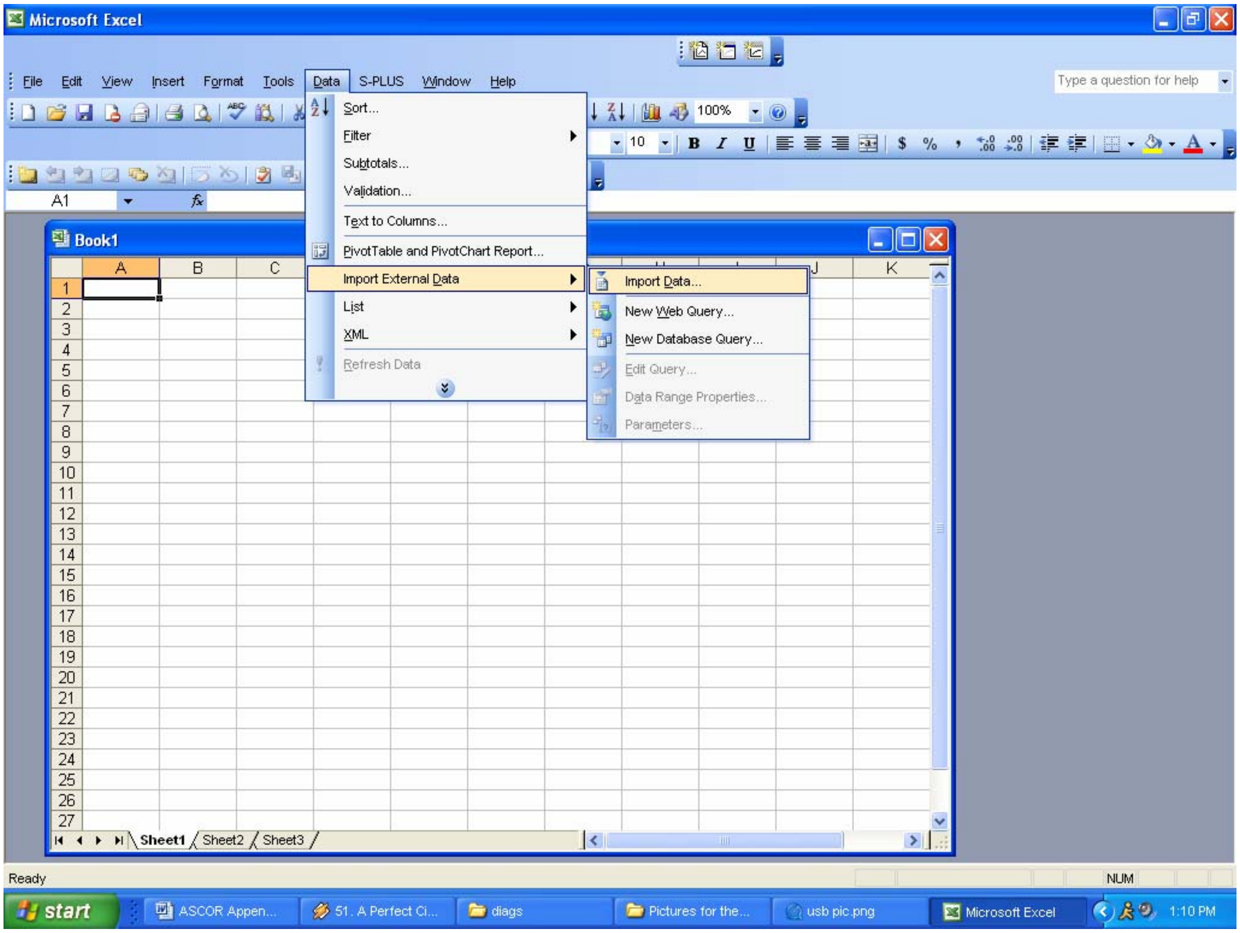


Figure C.12

Figure C.12 shows the options that need to be selected to begin the import process of ASCORE files into Excel.

Once this is done navigate to where the file is stored on the computer. Ensure that you have selected **All Files** under **File Types** or the data will not be visible. Once this has been done the Excel text import wizard should appear. In step one you should select the delimited option under data type and select the row that you want to start the import at. It is often unnecessary to import the header information. Once this has been done next should be selected (see Figure C.13).

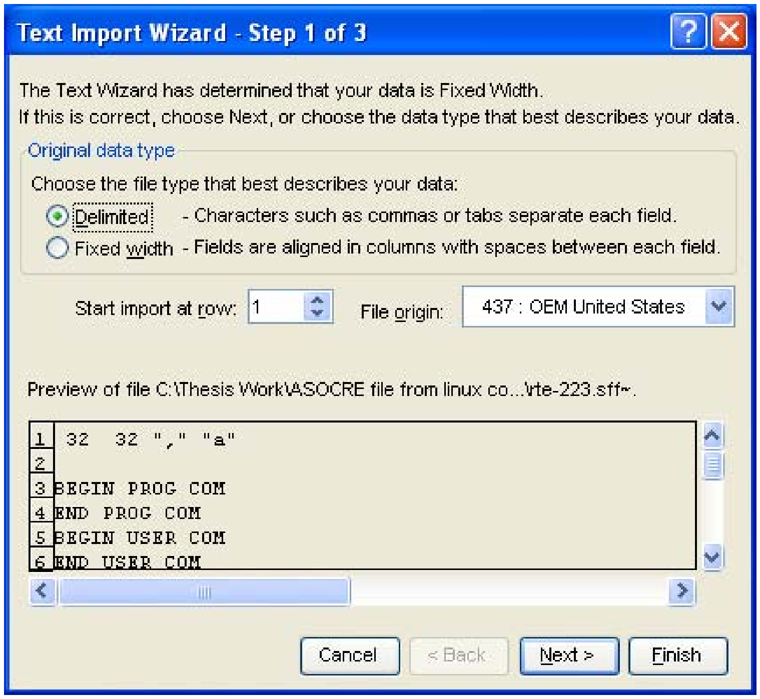


Figure C.13

Figure C.13 shows the set of options that need to be selected during the first step of the text import wizard.

At the next wizard, shown in Figure C.14, comma should be selected under delimiters and then the finish button.

Figure C.14

Figure C.14 shows the set of options that should be selected during the second step of the import text wizard.

This will open the file in Excel with the data aligned correctly by column. Once all changes have been made to the file, it should be saved as a CSV (comma delimited) file (see figure C.15).

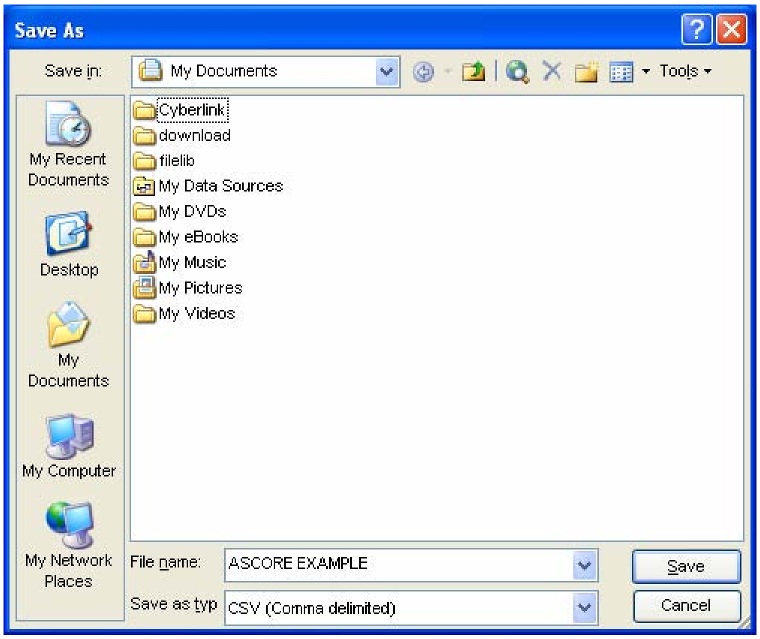


Figure C.15

Figure C.15 shows the options that need to be selected when saving an Excel file as a comma delimited CVS file.

This file can then be transferred back to the thumb drive and copied onto the Linux system. The final step in this process is getting the data from the CVS file into a readable and properly formatted file within ASCORE. This is a little tricky because the input files for ASCORE can only be modified once they have been opened with the kate command from the ASCORE command line window. The CVS file should be opened and the data that needs to be transferred over to ASCORE should be copied using either cntl-c or a right click. The file that needs to be modified should be opened using the kate command and then the data should be pasted into the file as appropriate and saved. It is always a good idea to create a backup copy of a file before changing it.

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX D: PRODAS VB SCRIPT: VONTEST

The VB Script below is an example of a general script written for a PRODAS simulation run for this thesis. A PRODAS script will automate simulation firings within PRODAS. The Vontest script runs a six degrees of freedom analysis and outputs a file with desired information. Vontest was used in this thesis as the base firing. The authors used S-Plus to generate many script files by using Vontest and randomizing desired parameters.

```
Sub Main

        'PRODAS Version 3 VB Script File 03/07/05

        'Map the path to projectile file
        testProj30="C:\ProgramFiles\ProdasV3\ProjectileData\MedCal\#30m789Cartridge.p
        r3"

        'Creates a projectile object named "proj30"
        set proj30=MacroSystem.InitializeProjectile

        'Opens an existing PRODAS data file
        proj30.openDataFile testProj30

        'Set Desired Parameters
                airplaneSetting=false
                airplaneAltitude=0.0
        terminationRange= 10000    'meters
                terminationAltitude= -100  'meters
                termiationTime= 3          'seconds
        quadrantElevation=-10.42   'degrees
                gunAzmith=0.0            'degrees
                muzzleVelocity=805.0     'meters per second
        initialxPos=-1000        'meters
                initialyPos=0                    'meters
                initialzPos=100          'meters
        metConditions=6          'environmental conditions: 1=cold, 2=polar,
             3=std,'4=tropical, 5=hot, 6=user
                atmosphereTemp=29.33     'degrees Celsius
                atmospherePress=986.56   'millibars
                windDirection=114.59     'degrees
                windSpeed=2.00                   'meters per second
        seaLevel=175.30                  'meters
                timeIncrement=.0009      'sec


        'The following code sets the single input parameters into PRODAS

        'AIRPLANE SETTINGS, this set of parameters applies for a moving aircraft.  If using ‘a stationary
        aircraft, skip...

        'Sets the option for using the integrated airplane settings in PRODAS
        ‘ true=yes
                proj30.setDataPointValue "Airplane","useAirplane",airplaneSetting

        'Sets the altitude of the airplane when projectile is fired
                proj30.setDataPointValue "AeroStability","stabAltitude",airplaneAltitude

        'SIMULATION TERMINATION OPTIONS, termination is executed when the first condition is met

                'This value determines the stopping range for the simulation, integer input.
```

```
        proj30.setDataPointValue "Trajectory", "RangeFinal",terminationRange

'This value determines the stopping altitude for the projectile, integer
 input.
        proj30.setDataPointValue "Trajectory" , "AltitudeFinal", terminationAltitude

        'This value determines the stopping time for the simulation
        proj30.setDataPointValue "Trajectory", "TimeFinal",termiationTime

'GUN SETUP

        'Quadrant elevation
        proj30.setDataPointValue "Trajectory","QE",quadrantElevation*3.141592654/180

        'Gun azimuth
        'proj30.setDataPointValue "Trajectory","Azimuth",gunAzmith*3.141592654/180

        'MuzzleVelocity
        proj30.setDataPointValue"metConditionsTrajectory","MuzzleVelocity",
        muzzleVelocity

'METEROLOGICAL CONDITIONS

        'environmental conditions: 1=cold, 2=polar, 3=std, 4=tropical, 5=hot, 6=user
        proj30.setDataPointValue "Met","metType",metConditions

        'Set Atmosphere Temperature
        proj30.setDataPointValue "MET","Temp at Sea Level",atmosphereTemp

        'Set Atmosphere Pressure
        proj30.setDataPointValue "MET","Pres at Sea Level",atmospherePress

        'Set Wind Direction
        proj30.setDataPointValue "MET","Wind Direction",windDirection*3.141592654/180

        'Set Wind Speed
proj30.setDataPointValue "MET","Wind Speed",windSpeed

        'Set Meters above Sea Level where measurements were taken
        proj30.setDataPointValue "MET","Altitude of Measurement",seaLevel

'PROJECTILE INITIAL LOCATION

        proj30.setDataPointValue "Trajectory","initialzposition",initialzPos

        proj30.setDataPointValue "Trajectory","initialyposition",initialyPos

        proj30.setDataPointValue "Trajectory","initialXposition",initialxPos

'OUTPUT SETUP

        proj30.setDataPointValue "Trajectory","plotOutIncrTime",timeIncrement

'Executes a MET2000 analysis.  This analysis calculates the meteorological ‘conditions with the
user inputs.

        proj30.executeAnalysis "MET2000"

'Executes a 6dof analysis.  This runs the firing simulation

        proj30.executeAnalysis "Traj20006D"

'Creates output text file
        set results=MacroSystem.InitializeResultsFile

results.openFile "C:\Program Files\ProdasV3\scripts\Script
Outputfile\FILE0001.txt"

results.writeHeader

results.writeString "PARAMETERS:"
```

```
proj30.writeResultsDataPoint "Trajectory","QE"
proj30.writeResultsDataPoint "MET","Wind Speed"
proj30.writeResultsDataPoint "MET","Wind Direction"
proj30.writeResultsDataPoint "MET","Pres at Sea Level"
proj30.writeResultsDataPoint "MET","Temp at Sea Level"
proj30.writeResultsDataPoint "Trajectory","initialxposition"
        proj30.writeResultsDataPoint "Trajectory","initialyposition"
proj30.writeResultsDataPoint "Trajectory","initialzposition"
proj30.writeResultsDataTable "Trajectory","TrajResultsData"
results.closeFile
end sub
```

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX E: S-PLUS FUNCTION: JONF

The S-Plus function “jonF” was created to perform several tasks which assisted in this thesis. This function was primarily written by Professor Samuel E. Buttrey. The function “jonF” reads a generic VB script named vontest and pulls desired parameters from the script. S-Plus then randomizes these parameters with a uniform distribution and then creates a new VB script using the new values of the parameters. The maximum number of scripts jonf can create is 1000 and each script is named “Scrp0001” to “Scrp1000.” Corresponding to each script, the output file PRODAS will create when the scripts is run will be named “FILE0001” to “FILE1000.” To run “jonF” on the S-Plus command line, type jonf(n), where n is the number of scripts desired.

```
function(n = 1000)
{
      # S-Plus reads & formats data from the VB script vontest
      infile <- scan(
            "c:/program files/prodasv3/scripts/vontest.pvb",sep = "\n")
            infile <- paste(infile, "\n", sep = "")
      # Temporarily specifies the directory where the scripts will
        output to
      #
      out <- "c:/program files/prodasv3/scripts/scrp"
      # Temporarily sets up the naming of the PRODAS output file
      res.line.head <- "\tresults.openFile \"C:\\ProgramFiles\\
      ProdasV3\\scripts\\Script Output file\\"
      # Searches in “infile” for the line which contains “testProj30”
      tp30 <- (1:length(infile))[regexpr("testProj30=", infile
            ) > 0]
      # Inserts the following directory for tp30
      infile[tp30] <- "testProj30=\"C:\\Program Files\\ProdasV3\\
      ProjectileData\\MedCal\\#30m789Cartridge.pr3\""
      # S-Plus searches for each parameter in infile
      qe <- (1:length(infile))[regexpr("quadrantElevation=",infile) >0]
      ix <- (1:length(infile))[regexpr("initialxPos=", infile) > 0]
      iy <- (1:length(infile))[regexpr("initialyPos=", infile) > 0]
      iz <- (1:length(infile))[regexpr("initialzPos=", infile) > 0]
      res <- (1:length(infile))[regexpr("results.openFile", infile) >0]
      for(i in 1:n) {
            # A value for each parameter is uniformly assigned to each
              parameter
            #
      new.qe <- runif(1, -7.28, -6.54)
            new.ix <- runif(1, -751, -739)
            new.iy <- runif(1, -6, 6)
            new.iz <- runif(1, 94, 106)
      if(i < 10)
                  xxxx <- paste("000", i, sep = "")
```

```
            else if(i < 100)
                  xxxx <- paste("00", i, sep = "")
            else if(i < 1000)
                  xxxx <- paste("0", i, sep = "")
            else xxxx <- i
            out.for.now <- paste(out, xxxx, ".pvb", sep = ""
                  )
            res.for.now <- paste(res.line.head, "FILE", xxxx,
            ".txt\"\n", sep = "")
            # The new value for each parameter is assigned in a new
              Script
            #
            infile[qe] <- paste("\tquadrantElevation=", round(new.qe,
            3), "\n", sep = "")
            infile[ix] <- paste("\tinitialxPos=", round(new.ix, 0),
            "\n", sep = "")
            infile[iy] <- paste("\tinitialyPos=", round(new.iy, 0),
            "\n", sep = "")
            infile[iz] <- paste("\tinitialzPos=", round(new.iz, 0),
            "\n", sep = "")
            infile[res] <- res.for.now
            cat(infile, file = out.for.now)
      }
}
```

APPENDIX F: SPLUS FUNCTION: PRODASPARAMS

The S-Plus function “prodasparams” was created to extract information about parameters for each simulation run and place all these parameters in a dataframe. This function was primarily written by Professor Samuel E. Buttrey. Specifically, prodasparams reads information from a PRODAS output file and makes a dataframe with each column being one of the following prameters: name of the output file, quadrant elevation, wind speed, wind direction, air pressure, air temperature, initial x position, initial y-position, and initial z-position.

```
function()
{
	# Designates the directory S-Plus is extracting information from
	base <- "C:\\Program Files\\ProdasV3\\scripts\\Script Output
	file\\"
	# Formats the naming convention for each PRODAS output file
	num <- paste(1:1000)
	num[nchar(num) == 1] <- paste("000", num[nchar(num) ==
		1], sep = "")
	num[nchar(num) == 2] <- paste("00", num[nchar(num) ==
		2], sep = "")
	num[nchar(num) == 3] <- paste("0", num[nchar(num) == 3],
		sep = "")
	# Creates a vector of the PRODAS output files (character string)
	file.name <- paste("FILE", num, ".txt", sep = "")
	# Creates a vector of PRODAS output files that are in base
	nm <- paste(base, file.name, sep = "")
	# Test for the existence of the PRODAS files in nm
	is.it.there <- file.exists(nm)
	# Checks to see if any PRODAS output files are in nm
	if(all(!is.it.there))
		stop("No <filennnn>'s found.")
	# Only selects the PRODAS output files that are in nm
	nm <- nm[is.it.there]
	file.name <- file.name[is.it.there]
	# Creates a matrix of dimensions length(nm) by 8
	out <- matrix(0, length(nm), 8)
	# List the names for the columns of matrix “out”
	dimnames(out) <- list(file.name, c("TRAJ.QE", "WINDSP",
		"wINDDIR", "PRESS", "TEMP", "INIT.X", "INIT.Y",
		"INIT.Z"))
	for(i in 1:length(nm)) {
		# In the PRODAS output file, rows 4-11 are selected
		gimme <- scan(nm[i], sep = "\n", n = 11)[ - (
			1:3)]
		# Out of rows 4-11, only characters to the right of the “=”
		  sign are selected and designated as numerical values
		#
		out[i,  ] <- as.numeric(unpaste(gimme, "=")[[
			2]])
	}
	out
}
```

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX G: S-PLUS FUNCTION: EXTRACT

The S-Plus function “extract” was created to perform several functions. This function was primarily written by Professor Samuel E. Buttrey. Extract reads data from a PRODAS output files. The first function extract serves is to find the location and time of VT impact. Extract also reports whether or not the round hit the target and the impact time and location. If the round hit the target then the impact time and location is impact on target. If the round missed the target then the impact time and location is impact on the ground.

```
function(n = 2)
{
      # Designates the directory S-Plus is extracting information from
        base <- "C:\\Program Files\\ProdasV3\\scripts\\Script Output
        file\\"
      # Formats the naming convention for each PRODAS output file
      n <- paste(n)
      n[nchar(n) == 1] <- paste("000", n[nchar(n) == 1], sep
             = "")
      n[nchar(n) == 2] <- paste("00", n[nchar(n) == 2], sep =
            "")
      n[nchar(n) == 3] <- paste("0", n[nchar(n) == 3], sep =
            "")
      file <- paste("FILE", n, ".txt", sep = "")
      pos <- c("X", "Y", "Z")
      a.buttload.of.zeros <- rep(0, length(n))
      # A dataframe named “output” is created and is initially filled
       with zero’s as inputs
      #
      output <- data.frame(File = file, Time = a.buttload.of.zeros)
      output$Z <- output$Y <- output$X <- a.buttload.of.zeros
      output$Hit <- rep(F, nrow(output))
      output$YR <- output$YR <- output$XR <- output$TimeR <
      a.buttload.of.zeros
      for(i in 1:length(n)) {
            x <- read.table(paste(base, file[i], sep = ""),
                  skip = 12, header = T, row.names = NULL)
            # [i]
            # "v1" is the first row (columns X, Y, and Z).
              "V2" is all the rows (hence the uppercase)
            #
            v1 <- t(as.matrix(x[1, pos]))
            V2 <- as.matrix(x[, pos])
            lv1 <- sqrt(sum(v1^2))
            lV2 <- sqrt(apply(V2^2, 1, sum))
            # dot is a matrix of angles between v1 and each V2
              dot <- (V2 %*% v1)/(lv1 * lV2)
            # dot.min finds the smallest absolute value of dot
            dot.min <- (1:nrow(x))[abs(dot) == min(abs(
```

```
            dot))][1]
        # Sets the output to be the corresponding X,Y,Z and Time
          that dot.min occurs.  This is the Virtual target impact
        #
        output[i, c("Time", "X", "Y", "Z")] <- x[dot.min,
            c("Time", "X", "Y", "Z")]
        # Dist2Tgt calculates the distance to the center of the
          target, (0,0,0), from each X,Y,Z position in the rounds
          trajectory
        #
        Dist2Tgt <- sqrt((x$X)^2 + (x$Y)^2 + (x$Z)^2)
        # If Dist2Tgt is less than 3 then the round is considered
          to hit the target.  Output now includes the time and
          location of target impact.
        #
        if(any(Dist2Tgt < 3.)) {
            output[i, "Hit"] <- T
            hitter <- (1:nrow(x))[Dist2Tgt < 3][
                1]
            output[i, c("TimeR", "XR", "YR", "ZR")] <-
                x[hitter, c("Time", "X", "Y",
                "Z")]
        }
        # If Dist2Tgt is never less than 3 then the round is
          considered to miss the target.  Output now includes the
          time and location of ground impact.
        #
        else {
            output[i, "Hit"] <- F
            min.z <- (1:nrow(x))[abs(x$Z) == min(
                abs(x$Z))][1]
            output[i, c("TimeR", "XR", "YR", "ZR")] <-
                x[min.z, c("Time", "X", "Y",
                "Z")]
        }
    }
    # "Output" is returned as a dataframe
    return(output)
}
```

LIST OF REFERENCES

1. Merriam-Webster Online Dictionary, Online Source: http://www.webster.com/cgi-bin/dictionary, May 2005.

2. Sundra, J., *Ballistics for Dummies*, Rifle Shooter, Online Source: http://www.rifleshootermag.com/shooting_tips/ballistics_0303/, June 2005.

3. *A Brief Course in Ballistics*, HorusVision, Online Source: http://www.horusvision.com/hv.cfm?pg=ballistics, June 2005.

4. Nennstiel, R., *How do bullets fly?*, Online Source: http://www.nennstiel-ruprecht.de/bullfly/, June 2005.

5. Army-Technology.com, Online Source: http://www.army-technology.com/glossary/muzzle-velocity.html, June 2005.

6. *PRODAS V3 User Manuel*, Arrow Tech Associates Inc., 2002.

7. Devore, J., *Probability and Statistics for Engineering and the Sciences*, Fifth Edition, Brooks/Cole Publishing Company, 2000.

8. Montgomery, D., Peck, E., Vining, G., *Introduction to Linear Regression Analysis*, Third Edition, John Wiley & Sons, Inc., 2001.

9. Harrell, F., *Regression Modeling Strategies*, Springe-Verlag New York, Inc, 2002.

10. *PRODAS V3 Technical Manuel*, Arrow Tech Associates Inc., 2002.

11. *PRODAS V3 VB Script User Manuel*, Arrow Tech Associates Inc., 2002.

12. Clark, S., Donatis, A., *VBScript: Programmer's Reference*, Wrox Publishing Company, 1999

13. Lomax, P., Petrusha, R., *VBScript in a Nutshell*, Second Edition, O'Reilly Publishing Company, 2003.

THIS PAGE INTENTIONALLY LEFT BLANK

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center
   Ft. Belvoir, VA

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, CA

3. Professor Lyn Whitaker (OR/Wh)
   Department of Operations Research
   Naval Postgraduate School
   Monterey, CA

4. Professor Samuel Buttrey (OR/Bu)
   Department of Operations Research
   Naval Postgraduate School
   Monterey, CA

5. Commander
   U.S. Army Yuma Proving Grounds
   ATTN: CSTE-DTC-YP-YT-AC-EA
   Yuma, AZ

6. Professor Peter Andre
   Mathematics Department
   United States Naval Academy
   Annapolis, MD

7. Professor BrendanCunningham
   Economics Department
   United States Naval Academy
   Annapolis, MD

8. ENS Jon S. Francisco, Jr.
   Sandy Hook, VA

9. ENS Kristopher E. von Krueger
   Seattle, WA